WIDE AREA SEARCH AND ENGAGEMENT SIMULATION VALIDATION

THESIS

Michael J. Marlin, Captain, USAF

AFIT/GAE/ENY/07-M17

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/07-M17

WIDE AREA SEARCH AND ENGAGEMENT SIMULATION VALIDATION

THESIS

Presented to the Faculty

Department of Aeronautical and Astronautical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Michael J. Marlin, BS
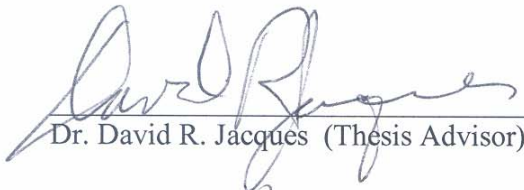
Captain, USAF

March 2007

AFIT/GAE/ENY/07-M17

WIDE AREA SEARCH AND ENGAGEMENT
SIMULATION VALIDATION

Michael J. Marlin, BS

Captain, USAF

Approved:

_____   16 MAR 07
Dr. David R. Jacques (Thesis Advisor)   Date

_____   March 15, 2007
Dr. Meir Pachter (Committee Member)   Date

_____   15 Mar 07
Maj Paul A. Blue (Committee Member)   Date

AFIT/GAE/ENY/07-M17

Abstract

The use of computer simulation in the development of cooperatively controlled unmanned combat aerial vehicles (UCAV) for autonomous wide area search and engagement applications is addressed. Computer simulation is an essential tool to analyze cooperative control algorithms designed to optimally employ multiple UCAVs in wide area search and engagement. To be representative of real world mission conditions a simulation must be able to accurately duplicate the performance of automatic target recognition (ATR) algorithms that will be used to discriminate between targets and non-targets in the battle space. The objective of this research is to demonstrate a method to validate a simulation's ATR model for future use in the evaluation of cooperative control schemes. This is accomplished by comparing the results from multiple simulations of academically contrived wide area search and engagement scenarios to closed form analytic solutions derived for the same scenarios.

Acknowledgments

Table of Contents

List of Figures

List of Tables

WIDE AREA SEARCH AND ENGAGEMENT SIMULATION VALIDATION

I. Introduction

Background

The Department of Defense (DoD) is interested in the development of low cost autonomous unmanned combat aerial vehicles (UCAV's) for use in wide area search and engagement applications. Warfighters envision a weapon system that can locate, identify and attack select targets in a given battle space with minimal human direction. Operational scenarios for this class of UCAV could include Theater Missile Defense (TMD), Suppression of Enemy Air Defense (SEAD), Persistent Area Denial (PAD), Urban Battlefield Surveillance (UBS) and any other application that involves searching defined geographic areas for specific target types.

One example of such a system is the Low Cost Autonomous Attack System (LOCAAS) shown in Figure 1. The LOCAAS is a concept demonstration of an autonomous wide area search and engagement system developed jointly by the Air Force Research Lab Munitions Directorate (AFRL/MN), Eglin AFB, Florida and Lockheed Martin Missiles and Fire Control, Orlando, Florida. It is a one-time-use wide area search munition (WASM) designed to autonomously locate and attack ground mobile targets with a multi-mode explosively formed penetrator warhead. The LOCAAS uses a Laser Radar (LADAR) seeker coupled with automatic target recognition (ATR) algorithms and an autonomous guidance and control system to detect, identify and engage enemy targets. It can be deployed from an aircraft or ground based launch system in groups or

individually with an expected endurance in the battle space of up to 30 minutes. Depending on mission objectives, the LOCAAS can be programmed to attack the first target it encounters or wait for a better target to come along before its endurance runs out. [14]

(Lockheed Martin photo)

Figure 1. Low Cost Autonomous Attack System (LOCAAS)

The LOCAAS ATR algorithms identify targets by comparing the LADAR seeker image of an encountered object to a preprogrammed library of target models. If the image is a quantified match to a target model the ATR will declare the object to be a target that can be attacked. Otherwise, the object is declared a false target that cannot be attacked. Although this is an oversimplification of the LOCAAS ATR, this description encapsulates the rudimentary ATR process that will be used in most autonomous wide area search and engagement systems.

The reliability of an ATR system to correctly classify encountered objects is of major concern. There always exists some probability that an ATR system will incorrectly classify an object. The two types of possible errors are a *false positive error* where a false target is declared a target, and a *false negative error* where a target is declared a false target. False positive errors will lead to false target attacks (i.e. collateral damage) and false negative errors infer lost opportunities for target attacks. The expected frequency of these errors must drive the development and operational use of any autonomous wide area search and engagement system. The goal of a wide area search and engagement system should be to maximize the number of target kills possible while limiting the number of false target attacks.

Previous Research

The Air Force Institute of Technology (AFIT) has produced a great deal of research on the wide area search and engagement problem over the past decade. The foundation of this body of research, in particular Jacques and Pachter [7], and subsequently Decker [1] and Kish [8], has dealt primarily with developing closed form analytic expressions for the expected outcomes (i.e. target and false target attacks) of various wide area search and engagement scenarios. Some of this analytic work, principally [7], has been applied in the Masters Theses of Gillen [4], Dunkel [3], Gozaydin [5] and Park [10] to appraise the benefits of cooperative behavior among multiple search agents. The common thread in their research was the use of computer simulations to analyze various cooperative behavior schemes. The Masters Thesis of Schulz [13] focused on validating the simulation tools used to evaluate wide area search

3

and engagement cooperative control algorithms. Cumulatively, this research body represents an advance in the underlying DoD goal of developing autonomous wide area search and engagement systems.

Jacques and Pachter [7] were the first at AFIT to derive analytic expressions to calculate the probabilities of target attack (P_{TA}) and false target attack (P_{FTA}) for multiple wide area search and engagement scenarios. They considered a single UCAV, armed with a single warhead performing an exhaustive and non-duplicative search of a battle space containing multiple targets and false targets. Their work examined six wide area search and engagement scenarios defined by different *a priori* distributions of targets and false targets. These scenarios, listed below, have been the basis for most of the AFIT research described in this section.

Scenario 1: A rectangular battle space region containing a single target with a uniform distribution and a Poisson field of false targets.

Scenario 2: Same battle space as Scenario 1 containing a Poisson field of targets and a Poisson field of false targets.

Scenario 3: Same battle space as Scenario 1 containing *N* targets with a uniform distribution and a Poisson field of false targets.

Scenario 4: Same battle space as Scenario 1 containing *N* targets and *M* false targets with uniform distributions.

Scenario 5: A circular battle space containing *N* targets with a circular normal distribution and a Poisson field of false targets.

Scenario 6: Same battle space as Scenario 5 containing N targets and M false targets with a circular normal distribution.

Decker [1] extended the work of Jacques and Pachter [7] by deriving analytic expressions for a multiple warhead capable UCAV for the six scenarios described above. He proposed methods for using his analytical expressions to evaluate cooperation schemes and rules of engagement for multiple search agents. Decker also suggested ways to formulate a control problem to optimally employ a wide area search and engagement UCAV using a selectable sensor threshold as the control parameter. He proposed that by manipulating sensor parameters as modeled by a Receiver Operating Characteristic (ROC) curve the expected P_{TA} for a system could be improved. A ROC curve is an experimentally determined relationship between sensor ATR probabilities for correct target declarations to false positive errors.

Kish [8] further developed a ROC curve based control problem formulation to maximize the P_{TA} subject to a constrained P_{FTA}. He developed a generalized dynamic optimization mathematical framework for all of the scenarios developed by Jacques and Pachter. His analytic expressions produce time varying optimal schedules of control input parameters for sensor gain settings and area coverage rates (i.e. UCAV velocity and altitude) during a mission.

Other research has applied the work of Jacques and Pachter [7] to evaluate cooperative behavior of multiple autonomous UCAVs in wide area search and engagement. In [6], Jacques provided some limited analysis of cooperative engagement of multiple single warhead UCAVs searching a common battle space. Jacques discussed

5

two classes of cooperative behavior: *cooperative engagement* and *cooperative classification*. Cooperative engagement is when one or more UCAVs attack a target that was detected and classified by another. Cooperative classification involves two or more UCAVs performing a sensor sweep of a target in order to classify it. Jacques also discussed the importance of false target attack rates and the distribution of targets as critical factors to the wide area search and engagement problem. For scenarios with high densities of false targets, cooperative engagement was potentially detrimental because the possibility of a false target attack becomes more likely before a real target is even encountered.

Some of Jacques' students expanded on this research. Dunkel [3] and Gozaydin [5] showed that cooperative behavior can decrease the occurrence of targets attacks and increase false target attacks. Their research considered cooperative and non-cooperative scenarios of multiple single warhead UCAVs searching a common battle space. Dunkel found that cooperative engagement was likely to result in a higher false target attack rate than non-cooperative behavior. He concluded that this was because there were typically several UCAVs ready to attack any false targets encountered that were classified as a targets. Dunkel did note that cooperative classification offered improved ATR declaration accuracy. Both Dunkel and Gozaydin found that the quantity of target attacks decreased with cooperative classification because extra care and time was spent classifying fewer encountered targets. The endurance of a group of UCAVs would simply run out before many targets could be classified and attacked.

Dunkel and Gozaydin used computer simulations to generate their results. The simulation they used was the MultiUAV [11] simulation environment developed by the Air Force Research Lab, Air Vehicles Directorate, Control Sciences Division (AFRL/VAC), Wright-Patterson AFB, Ohio. The simulation was written specifically to analyze cooperative control algorithms for autonomous wide area search and engagement applications. In early versions of MultiUAV, the probability for correctly classifying targets was dependent only on the number of sensor sweeps executed on a target by one or more simulated UCAVs. Correct classification was guaranteed provided the simulation time did not run out before a sufficient number of sensor sweeps could be performed. This did not adequately duplicate a realistic sensor/ATR system process performance because no consideration was made for target declaration errors.

In [3], Dunkel modified MultiUAV to allow false positive and false negative errors to occur in the simulation ATR process. His results compared favorably with the analytic work of Jacques and Pachter. Schulz took this one step farther and validated the MultiUAV simulation results against the analytical solutions for all six scenarios. Schulz's work was limited to a UCAV armed with a single warhead and constant ATR performance parameters. This research effort represents a similar validation of the MultiUAV simulation as the work of Schulz [13] except that it covers modifications to the MultiUAV simulation that allow for multiple warheads per UCAV as well as time varying ATR performance control parameters.

Research Objective

The primary goal of this research is to demonstrate a method to validate a generic simulation environment, such as MultiUAV, for use in the evaluation of cooperatively controlled autonomous UAVs in wide area search and engagement applications. More specific objectives are:

1. Empirically characterize the occurrence of target and false target attacks in MultiUAV given a single search agent with multiple warheads and compare these results to analytically derived solutions.

2. Evaluate the ability of MultiUAV to incorporate time varying parameters to optimize the simulated ATR performance for wide area search and engagement research applications.

3. Identify deficiencies (if any) in MultiUAV that are uncovered during this research effort.

Methodology and Scope

This work presents a method to validate the ability of a simulation environment such as MultiUAV to realistically duplicate the performance of ATR algorithms used to discriminate between targets and false targets in wide area search and engagement applications. This was accomplished by comparing the results from MultiUAV simulation runs of Scenario 1 and Scenario 2 to the analytic framework developed by Decker and Kish. Scenario 2 was used to evaluate the impact of multiple warheads on the performance of MultiUAV (Objective 1). Scenario 1 was used to evaluate the ability

to apply time varying parameters to optimally modify the performance of the ATR model

in MultiUAV (Objective 2). The following performance parameters were examined:

- Probability of target attack, P_{TA}

- Probability of false target attack, P_{FTA}

- Expected number of target attacks, E_{TA}

- Expected number of false target attacks, E_{FTA}

This research does not consider the case of multiple UCAVs in the battle space or

evaluate the performance of any specific UCAV design concepts or wide area search and

engagement applications. Instead, this work provides a baseline validation and analysis

of the MultiUAV simulation environment for future use in wide area search and

engagement research applications (Objective 3).

Overview of Thesis

The following chapter, Chapter 2, further develops the wide area search and

engagement problem setup including definitions for the battle space region and the

distribution of targets and false targets in the battle space. The fundamental analytic

theory for calculating P_{TA} and P_{FTA} as well as the formulation of the basic control

problem are also provided. Chapter 3 gives a general description of the MultiUAV

simulation environment and the required setup and modifications to MultiUAV specific

to this research. Chapter 4 will present the empirical results from various MultiUAV

simulations that were run to accomplish the research objectives and how these results

compare to the analytic theory from Chapter 2. Finally, Chapter 5 provides the

conclusions of this work and recommendations for further research.

II. Analytic Background for Wide Area Search and Engagement

Chapter Overview

Potential uses for wide area search and engagement systems could include Theater Missile Defense (TMD), Suppression of Enemy Air Defense (SEAD), Persistent Area Denial (PAD), Urban Battlefield Surveillance (UBS) and many other applications that involve searching defined geographic areas for specific targets. These mission types can differ greatly in terms of the size of the battle space and density of targets and false targets in that battle space. For example, a TMD scenario may consist of a large search area covering hundreds of square miles containing only a few targets and false targets. In contrast, a UBS scenario might be restricted to a few city blocks with one or two potential targets mixed in a high-density field of false targets. To construct a computer simulation that models these types of scenarios the battle space and target densities need to be well defined. Validation of simulation results may then be accomplished through direct comparison with analytic solutions developed for each scenario.

This chapter describes the generic battle space and distributions of targets and false targets used to model wide area search and engagement applications. In addition, the chapter defines the performance of an ATR algorithm in terms of the probability of correct target classifications. From this, a brief overview is given of the fundamental analytic theory for calculating P_{TA} and P_{FTA} for the academic Scenarios 1 and 2. Finally the basic control problem formulation is developed by modeling ATR performance in terms of a ROC curve for the single warhead UCAV case of Scenario 1.

Battle Space Description

Jacques' and Pachter's [7] six wide area search and engagement scenarios utilized either a rectangular (Scenarios 1-4) or circular (Scenarios 5 and 6) battle space region. This work is confined to Scenarios 1 and 2; therefore, only a rectangular battle space as shown in Figure 2 is considered.
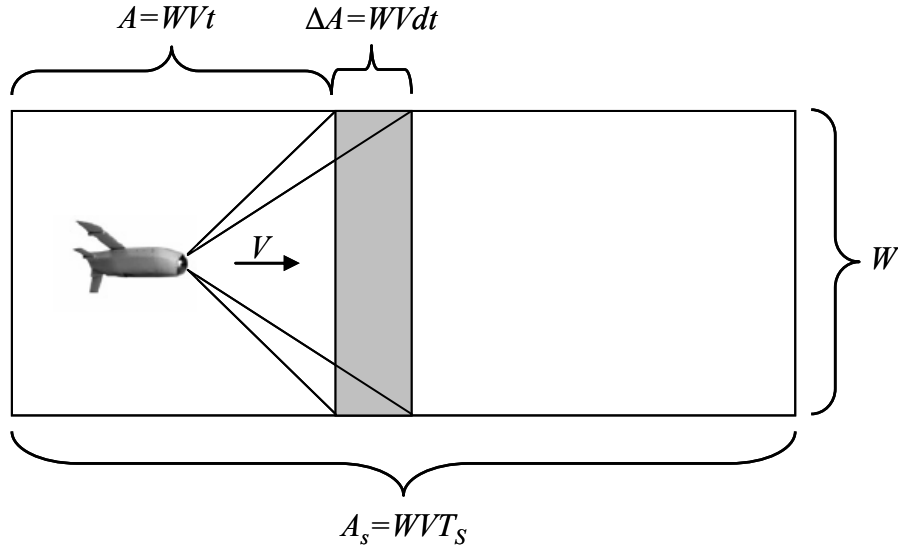
Figure 2. Battle Space Diagram

From Figure 2, the UCAV searches the regions A_S with area A_S at a constant altitude and velocity V over some total time interval T_S. The area that has been searched up to time t, $0 < t \leq T_S$, is denoted as A. The sensor footprint of the UCAV can be represented by an incremental area ΔA defined by velocity V, the sensor swath width W, and some time step dt. It is convenient to normalize time t and the area searched A to the variable x, $0 \leq x \leq 1$, where

$$x = \frac{t}{T_s} \,, \qquad (1)$$

and

$$A = A_S x \,. \qquad (2)$$

Distribution of Targets

For Scenario 1, region A_S contains a single target with a uniform distribution and a Poisson field of false targets. As a property of the uniform distribution the location of the single target in A_S can be said to be completely random. The probability that a search agent encounters the target in any arbitrary sub-region of A_S defined as having area A is

$$P(\text{Target in A}) = \frac{A}{A_s} \,. \qquad (3)$$

The Poisson field of false targets in A_S is characterized by a probability density parameter α which is the expected number of false target encounters per unit area. The Poisson probability parameter λ_{FT}, or expected quantity of false targets in area A, is then

$$\lambda_{FT} = \alpha A \,, \qquad (4)$$

and the probability of encountering exactly k false targets in area A is defined as

$$P(FT = k) = e^{-\lambda_{FT}} \frac{\lambda_{FT}^k}{k!} \,, \quad k = 0,1,2... \qquad (5)$$

For Scenario 2, A_S contains a Poisson field of targets and a Poisson field of false targets. The Poisson field of false targets remains the same as in Scenario 1. The Poisson field of targets is characterized by a probability density parameter β which is the

expected number of target encounters per unit area. The Poisson probability parameter λ_T, is then

$$\lambda_T = \beta A, \qquad (6)$$

and the probability of encountering exactly t targets in A becomes

$$P(T = t) = e^{-\lambda_T} \frac{\lambda_T^t}{t!}, \quad t = 0,1,2... \qquad (7)$$

Probability of Target Report

As a UCAV searches A_S it will start to encounter targets and false targets and execute an ATR process to classify them. The probability of the ATR process correctly reporting an encountered target as a target can be denoted as P_{TR} and the probability of incorrectly reporting a target is $(1-P_{TR})$. The comparable notation for the ATR declaration of an encountered false target is P_{FTR} and $(1-P_{FTR})$.

Confusion Matrix

For the simple case of a single target type, Jacques and Pachter [7] modeled the ATR process in terms of a binary *confusion matrix,* shown in Table 1.

Object Reported As:		Object Encountered	
		T	FT
	T	P_{TR}	$1-P_{FTR}$
	FT	$1-P_{TR}$	P_{FTR}

Table 1. Basic Confusion Matrix

13

Ideally the confusion matrix will take the form of the identity matrix where $P_{TR} = P_{FTR} = 1$. This infers no incorrect target declarations can occur. It is more realistic to define the values for P_{TR} as $0 < P_{TR} \leq 1$ and P_{FTR} as $0 < P_{FTR} \leq 1$ to account for the likelihood of target declaration errors. Overall, the confusion matrix implies there are three possible wide area search and engagement outcomes for any given ATR report:

- A target is encountered and declared a target resulting in a Target Attack (*TA*).

- A false target is encountered and declared a target resulting in a False Target Attack (*FTA*).

- A false target is encountered and declared a false target or a target is encountered and declared a false target resulting in no attack.

Probability of Target Attack

Scenario 1: Single Warhead UCAV

For the single warhead UCAV case of Scenario 1, the probability of encountering a target in the sensor footprint ΔA is predicated on the probability that the target is located in ΔA and the probability that no false target attack occurred previously in area A. The probability of a target encounter can be written as

$$P_{TE}(\Delta A) = \frac{\Delta A}{A_S} P_{\overline{FTA}}(A) \qquad (8)$$

where $P_{\overline{FTA}}(A)$ represents the probability of no false target attacks occurring prior to arriving at area ΔA. The probability of a target attack is then a matter of an encountered

14

target being correctly reported as represented by P_{TR}. P_{TR} is either experimentally determined or can be arbitrarily defined. Equation 8 becomes

$$P_{TA}(\Delta A) = \frac{\Delta A}{A_S} P_{\overline{FTA}}(A) P_{TR}.$$

(9)

Jacques and Pachter [7] showed that for a Poisson field of false targets $P_{\overline{FTA}}(A)$ is the simple case of $k=0$ for Equation 5 and can be rewritten as

$$P_{\overline{FTA}}(A) = e^{-(1-P_{FTR})\alpha A}.$$

(10)

Substituting Equation 10 into Equation 9 gives

$$P_{TA}(\Delta A) = \frac{\Delta A}{A_S} P_{TR} e^{-(1-P_{FTR})\alpha A}.$$

(11)

Let $f(t)$ be the probability density function (p.d.f.) for a target attack event occurring at time t, $0 < t \le T_S$ and $f(t)dt$ be the probability of an attack occurring over the time interval $[t, t+dt]$. Also, recall from Figure 2 that $\Delta A = WVdt$. Equation 11 can then be rewritten as

$$f(t)dt = \frac{WVdt}{A_S} P_{TR} e^{-(1-P_{FTR})\alpha A}.$$

(12)

The p.d.f for a target attack event can now be written as

$$f(t) = \frac{WV}{A_S} P_{TR} e^{-(1-P_{FTR})\alpha A}.$$

(13)

Since $A_S = WVT_S$ and $A = WVt$ then

$$A = A_S \frac{t}{T_S},$$

(14)

15

and

$$\frac{WV}{A_S} = \frac{1}{T_S}.$$ (15)

From Equation 4 the Poisson parameter for area A_S is

$$\lambda_{FT} = \alpha A_S,$$ (16)

and

$$\alpha A = \lambda_{FT} \frac{t}{T_S}.$$ (17)

Substitute Equations 15 and 17 into Equation 13 and the target attack *p.d.f.*, can be rewritten as

$$f(t) = \frac{1}{T_S} P_{TR} e^{-(1-P_{FTR})\lambda_{FT} \frac{t}{T_S}}.$$ (18)

In a similar process, Jacques and Pachter [7] derived the *p.d.f.* for a false target attack event at time τ, $0 < \tau \le T_S$ as

$$g(\tau) = \frac{1}{T_S} \lambda_{FT} \left(1 - P_{FTR}\right) \left(1 - P_{TR} \frac{\tau}{T_S}\right) e^{-(1-P_{FTR})\lambda_{FT} \frac{\tau}{T_S}}.$$ (19)

The probability of a target or false target attack occurring in A_S can be written as the integral of the *p.d.f.* through the whole battle space or

$$P_{TA} = \int_0^{T_S} f(t)dt \text{ and } P_{FTA} = \int_0^{T_S} g(\tau)d\tau.$$

Integrating Equations 18 and 19 gives the closed form analytic expressions for P_{TA} and P_{FTA} as

$$P_{TA} = P_{TR} \frac{1 - e^{-(1-P_{FTR})\lambda_{FT}}}{(1-P_{FTR})\lambda_{FT}}, \qquad (20)$$

and

$$P_{FTA} = \left[1 - \frac{P_{TR}}{(1-P_{FTR})\lambda_{FT}}\right]\left[1 - e^{-(1-P_{FTR})\lambda_{FT}}\right] + P_{TR}e^{-(1-P_{FTR})\lambda_{FT}}. \qquad (21)$$

The probability of no attack occurring is then

$$1 - (P_{TA} + P_{FTA}) = (1-P_{TR})e^{-(1-P_{FTR})\lambda_{FT}}. \qquad (22)$$

Scenario 2: Single Warhead UCAV

For the single warhead UCAV case of Scenario 2, Jacques and Pachter [7] derived the *p.d.f.* equations for target attack and false target attack as

$$f(t) = \frac{1}{T_S} P_{TR}\lambda_T e^{-\left[(1-P_{FTR})\lambda_{FT} + P_{TR}\lambda_T\right]\frac{t}{T_S}} \qquad (23)$$

and

$$g(\tau) = \frac{1}{T_S}(1-P_{FTR})\lambda_{FT} e^{-\left[(1-P_{FTR})\lambda_{FT} + P_{TR}\lambda_T\right]\frac{\tau}{T_S}}. \qquad (24)$$

Integrating Equations 23 and 24 over A_S gives P_{TA} and P_{FTA} as

$$P_{TA} = \left[\frac{P_{TR}\lambda_T}{(1-P_{FTR})\lambda_{FT} + P_{TR}\lambda_T}\right]\left[1 - e^{-\left[(1-P_{FTR})\lambda_{FT} + P_{TR}\lambda_T\right]}\right], \qquad (25)$$

and

$$P_{FTA} = \left[\frac{(-P_{FTR})\lambda_{FT}}{(1-P_{FTR})\lambda_{FT} + P_{TR}\lambda_T}\right]\left[1 - e^{-\left[(1-P_{FTR})\lambda_{FT} + P_{TR}\lambda_T\right]}\right]. \qquad (26)$$

Scenario 2: Multi Warhead UCAV

Decker [1] expanded on Jacques' and Pachter's equations to consider a multiple warhead UCAV. He developed expressions for the probabilities of a specific number of target and false target attacks given a single UCAV searching A_S armed with w warheads. In these expressions Decker utilized a Poisson parameters for the density of target and false target attacks defined

$$\lambda_{TA} = P_{TR}\lambda_T ,$$

and

$$\lambda_{FTA} = \left(1 - P_{FTR}\right)\lambda_{FT} .$$

Decker's expressions for P_{TA} and P_{FTA} are written as

$$P_{TA=t|w} = \sum_{t=0}^{w-t-1}\left[\frac{\left(\lambda_{TA}x\right)^t}{t!}\frac{\left(\lambda_{FTA}x\right)^f}{f!}e^{-\left(\lambda_{TA}+\lambda_{FTA}\right)x}\right] +$$

$$\lambda_{TA}^t\lambda_{FTA}^{w-t}\binom{w}{t}\frac{1}{\left(\lambda_{TA}+\lambda_{FTA}\right)^w}\frac{\gamma\left(w,\left(\lambda_{TA}+\lambda_{FTA}\right)x\right)}{\Gamma\left(w\right)}, \qquad (27)$$

and

$$P_{FTA=f|w} = \sum_{t=0}^{w-f-1}\left[\frac{\left(\lambda_{TA}x\right)^t}{t!}\frac{\left(\lambda_{FTA}x\right)^f}{f!}e^{-\left(\lambda_{TA}+\lambda_{FTA}\right)x}\right] +$$

$$\frac{\lambda_{TA}^{w-f}\lambda_{FTA}^f}{\left(\lambda_{TA}+\lambda_{FTA}\right)^w}\binom{w}{f}\frac{\gamma\left(w,\left(\lambda_{TA}+\lambda_{FTA}\right)x\right)}{\Gamma\left(w\right)}. \qquad (28)$$

18

In Equations 27 and 28 above $\Gamma(w)$ is the Gamma function which can be written as

$$\Gamma(w) = (w-1)!,$$

and $\gamma(w,(\lambda_{TA}+\lambda_{FTA})x)$ is the incomplete Gamma function which can be written as

$$\gamma\left(w,\left(\lambda_{TA}+\lambda_{FTA}\right)x\right) = \int_{w}^{(\lambda_{TA}+\lambda_{FTA})x} e^{-\zeta}\zeta^{w-1}d\zeta .$$

Decker also derived equations for the expected number of target and false target attacks as

$$E[t] = \sum_{f=0}^{w}\left[t\frac{\lambda_{TA}^{t}\lambda_{FTA}^{w-t}}{\left(\lambda_{TA}+\lambda_{FTA}\right)^{w}}\binom{w}{t}\frac{\gamma\left(w,\left(\lambda_{TA}+\lambda_{FTA}\right)x\right)}{\Gamma(w)}\right] + \sum_{f=0}^{w-1}\sum_{t=0}^{w-t-1}\left[t\frac{\left(\lambda_{TA}x\right)^{t}}{t!}\frac{\left(\lambda_{FTA}x\right)^{f}}{f!}e^{-(\lambda_{TA}+\lambda_{FTA})x}\right] \qquad (29)$$

and

$$E[f] = \sum_{f=0}^{w}\left[f\frac{\lambda_{TA}^{w-f}\lambda_{FTA}^{f}}{\left(\lambda_{TA}+\lambda_{FTA}\right)^{w}}\binom{w}{f}\frac{\gamma\left(w,\left(\lambda_{TA}+\lambda_{FTA}\right)x\right)}{\Gamma(w)}\right] + \sum_{f=0}^{w-1}\sum_{t=0}^{w-f-1}\left[f\frac{\left(\lambda_{TA}x\right)^{t}}{t!}\frac{\left(\lambda_{FTA}x\right)^{f}}{f!}e^{-(\lambda_{TA}+\lambda_{FTA})x}\right] \qquad (30)$$

Control Problem Formulation

Receiver Operating Characteristic

Before the control problem formulation can be presented, the concept of a *Receiver Operating Characteristic* (ROC) must be discussed. The control inputs used in the control problem formulation are adapted from a ROC curve. It is widely

acknowledged in sensor signal processing theory that the probabilities for correct target classifications and false positive errors [i.e. P_{TR} and ($1-P_{FTR}$)] are not independent and can be modeled in terms of a ROC curve. Although the ROC curve is determined experimentally, a mathematical model of the curve can be used analytically relate P_{TR} and ($1-P_{FTR}$). Kish [8] adapted the following ROC curve equation as a model for a generic wide area search and engagement sensor/ATR system.

$$\left(1-P_{FTR}\right)=\frac{P_{TR}}{\left(1-c\right)P_{TR}+c} \tag{31}$$

A plot of both Kish's and Decker's ROC curve equations with P_{TR} on the y-axis and ($1-P_{FTR}$) on the x-axis will starts at (0, 0) and monotonically increase to (1, 1). Figure 3 shows a family of ROC curves as defined by Equation (31).
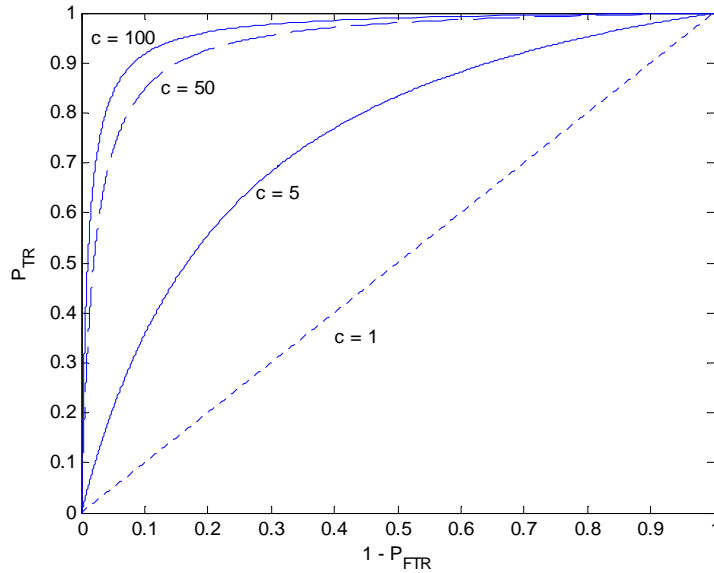
Figure 3. Family of ROC Curves

Decker [1] used an alternate ROC curve equation with a sensor performance parameter q.

$$P_{FTR} = 1 - P_{TR}^q \qquad (32)$$

In Equation 31, the scalar parameter c, $1 < c < \infty$, (or q in Equation 32) is dependent on factors such as UCAV velocity and altitude. It may also be limited by the quality of the sensor, performance of the ATR algorithms or even aspect angle of the sensor to the target. The value of c determines which curve in a family of ROC curves characterizes a sensor/ATR system's performance. As can be seen in Figure 3, larger values for c (or q) allow for higher values of P_{TR} before ($1 - P_{FTR}$) undergoes any significant rate of increase. Kish proposed that by varying UCAV velocity and altitude, c can be utilized as a time varying control input to optimize ATR performance. In both Equations 31 and 32, the value of P_{TR} determines the specific position on a given ROC curve that a sensot/ATR system operates from relative to ($1 - P_{FTR}$). The parameter P_{TR} can then be used as a control input to strike a balance between P_{TA} and P_{FTA}. This is the basis for the control problem formulation.

Basic Problem Formulation

Kish [8] developed a generalized optimization control problem framework for each of Jacques' and Pachter's six academic scenarios that maximized P_{TA} subject to a constrained P_{FTA}. The basic optimization problem statement is as follows:

$$\text{Maximize:}_{c \text{ and/or } P_{TR}} \quad P_{TA}$$

$$\text{Subject to:} \quad P_{FTA} \leq P_{FTA_{max}}$$

The constraint, $P_{FTA_{max}}$ is set by system engineers or the warfighter. The ROC curve theory above provides the control inputs c which represents a UCAV's *area coverage rate* and P_{TR} which represents the system *sensor threshold*. Kish's control problem formulations used the following three control input cases:

- Fixed Area Coverage Rate, Fixed Sensor Threshold

- Fixed Area Coverage Rate, Variable Sensor Threshold

- Variable Area Coverage Rate, Variable Sensor Threshold

Rosario [12] adapted Kish's work into the following formulation for a discrete-time dynamic optimization problem for the single warhead UCAV case of Scenario 1. The control inputs are a fixed area coverage rate c and variable sensor threshold P_{TR}. The problem statement becomes

$$\underset{c \text{ and } P_{TR}(t)}{\text{Minimize:}} \quad \phi\big[s(N)\big] = -P_{TA}$$

$$\text{Subject to: } s(i+1) = f\big[s(i), u(i), i\big]$$
$$\psi\big[s(N)\big] = P_{FTA} \le P_{FTA_{max}}$$

where $s(i+1)$ represents a state vector of equations $\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix}$ and $u(i) = P_{TR}$.

From Equation 20 for P_{TA} the objective function becomes

$$\phi\big[x(N)\big] = \sum_{i=1}^{N} u(i-1) e^{-\lambda_{FT} x(i-1)} \Delta t \qquad (33)$$

From Equation 21 the constraint equation is

$$\psi\big[s(N)\big] = \sum_{i=1}^{N} \frac{\lambda_{FT} u(i-1)}{(1-c) u(i-1) + c} \big[1 - y(i-1)\big] e^{-\lambda_{FT} x(i-1)} \Delta t \qquad (34)$$

22

The state equations $x(i+1)$ and $y(i+1)$ are pulled from the *p.d.f.* Equations 18 and 19 for target and false target attacks through a series of mathematical manipulations. For greater detail on this derivation refer to [12]. For simplicity we can say the state equations are an integration of the ROC curve based values for P_{TR} and $(1-P_{FTR})$ over the region A_S written as

$$x(i+1) = x(i) + \frac{u(i)}{(1-c)u(i)+c}\Delta t, \qquad (35)$$

and

$$y(i+1) = y(i) + u(i)\Delta t. \qquad (36)$$

For comparison, Rosario also developed the control input case of a fixed area coverage rate and fixed sensor threshold. This problem formulation can be solved using a numerical software package such as Mathworks *Matlab®*.

Chapter Summary

This chapter developed the analytic background theory and expressions required to conduct a thorough validation of a wide area search and engagement computer simulation. The theory and expressions presented for Scenario 2, particularly Equations 27, 28, 29, and 30 will be used to validate the ability of MultiUAV to realistically simulate the performance of an ATR process with a multiple warhead capable UCAV (Research Objective 1). The control problem formulation developed by Rosario [12] as presented above will be used to evaluate the ability of MultiUAV to incorporate time varying parameters to dynamically optimize ATR performance (Research Objective 2).

23

Chapter 3 will provide a brief overview of the MultiUAV simulation environment, the key simulation functions used to model the ATR process, and modifications to the simulation required to implement the above academic scenarios to accomplish the research objectives.

III. MultiUAV Simulation Environment

Chapter Overview

Computer simulation is an essential tool for the research and development of cooperative control algorithms to optimally employ multiple UCAVs in wide area search and engagement. To be representative of real world conditions, a simulation must be able to accurately model the performance of the ATR algorithms that will be used to discriminate between targets and false targets. To validate this ability, the outcomes of basic wide area search and engagement scenarios implemented in a simulation environment like MultiUAV may be compared to the solutions from analytic expressions such as those presented in Chapter 2.

This chapter presents a basic overview of the core MultiUAV simulation environment and the embedded simulation functions most relevant to modeling a wide area search and engagement ATR process. These functions include: the battle space setup, how targets and false targets are distributed in the simulated battle space, the sequence of events from target discovery to kill, and most importantly the methodology used to classify encountered targets and false targets. In addition, this chapter describes modifications made to theses functions as well as some additions to MultiUAV to enable simulations of the academic Scenarios 1 and 2 as described in Chapter 2. These modifications include: the ability to accurately simulate Poisson fields of targets and false targets, the implementation of the confusion matrix and ROC curve equations to model

an ATR process and finally the use of time varying parameters as control inputs to optimize ATR performance.

MultiUAV Simulation Description

AFRL/VACA developed MultiUAV specifically to analyze cooperative control algorithms for use in autonomous wide area search and engagement applications. MultiUAV is built on the Mathworks *Matlab®* and *Simulink®* programming environments [11] as well as C++ functions. It is an open source simulation tool that has been adapted for use in numerous ongoing DoD, industry and academic research studies for cooperative behavior of autonomous vehicles.
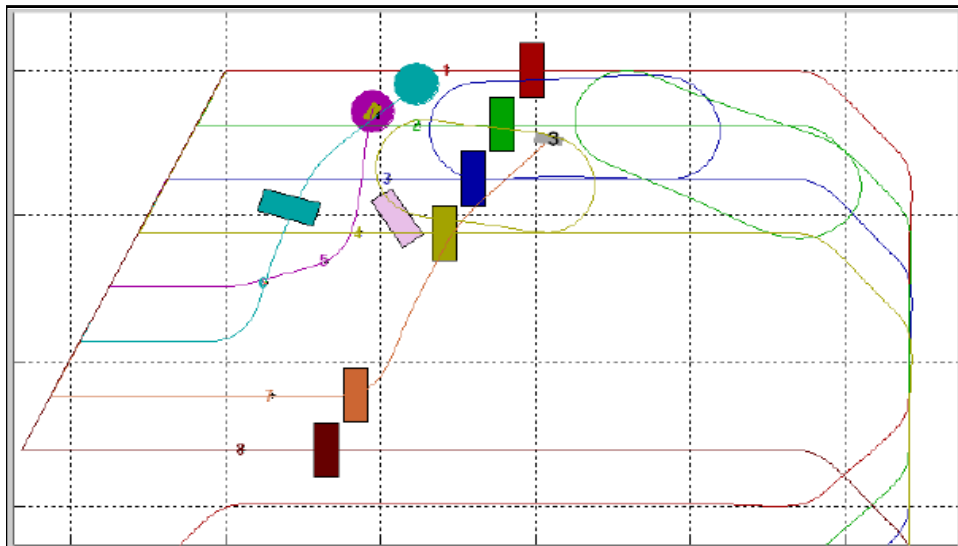
Figure 4. MultiUAV Simulation Output Plot

The basic simulation run consists of multiple UCAVs searching a defined area in a prescribed pattern for a given quantity/density of targets. The user can modify

simulation parameters such as quantity of UCAVs and warheads, arrangement and type of targets and false targets, and level of cooperative behavior. The typical MultiUAV simulation run will produce an output plot such as the one shown in Figure 4 [11], that shows the UCAV paths and target locations.

Simulation studies of specific scenario designs are executed using a Monte Carlo method to produce statistically significant results. Each Monte Carlo run uses an incrementing schedule of pseudo-random numbers to determine target locations and performance measures that can be compared to various threshold criteria.

Simulation Functions

For this research the MultiUAV functions of greatest concern involve how targets and false targets are distributed in the simulated battle space, the sequence of events from target discovery to kill, and most importantly the ATR process used to classify encountered targets and false targets is implemented.

Battle Space

The basic MultiUAV battle space is rectangular in shape with dimensions specified according to the desired scenario parameters. The battle space is searched in a back and forth "mowing the lawn" pattern. For this research the battle space illustrated in Figure 2 is employed. The width of the battle space is specified to be equal to the sensor swath width of the UCAV. The speed of the UCAV is held constant and the duration of all simulation runs are identical. The length of the battle space can be defined as the velocity of the UCAV multiplied by the total simulation run time. No UCAV

maneuvering is allowed. While this represents a greatly simplified simulation scenario, it will allow for direct comparison with the analytic expressions discussed in Chapter 2.

Target Distribution and Target Discovery Times

The MultiUAV simulation allows for targets to be arranged in the battle space according to a uniform, normal, or pseudo-Poisson distribution algorithm depending on the desired scenario parameters. Targets are inserted into the battle space by means of a numeric array of *target discovery times* that is generated using one of the distribution algorithms. In the case of the uniform distribution, the target discovery time array is generated using the Matlab random number generator command *RAND(N)* where *N* is the number of desired targets. The RAND(*N*) command produces an array of *N* random numbers with values between 0 and 1. The array is scaled to the simulation environment by simply multiplying each element in the array by the total simulation run time. As the simulation runs individual targets are placed in front of the sensor footprint of the UCAV according to the discovery times in the array. Therefore, the spatial coordinate of the targets are directly linked to that of the UCAV at the target discovery times. Because the UCAV moves through the simulated battle space in a straight line and constant speed this method accurately creates the desired target spatial distribution in the battle space.

Target State Machine and UCAV Task Assignments

As a UCAV searches the virtual battle space it will start to encounter targets. The target state machine shown in Figure 5 [11] illustrates the target "kill chain" from target detection to verified kill.
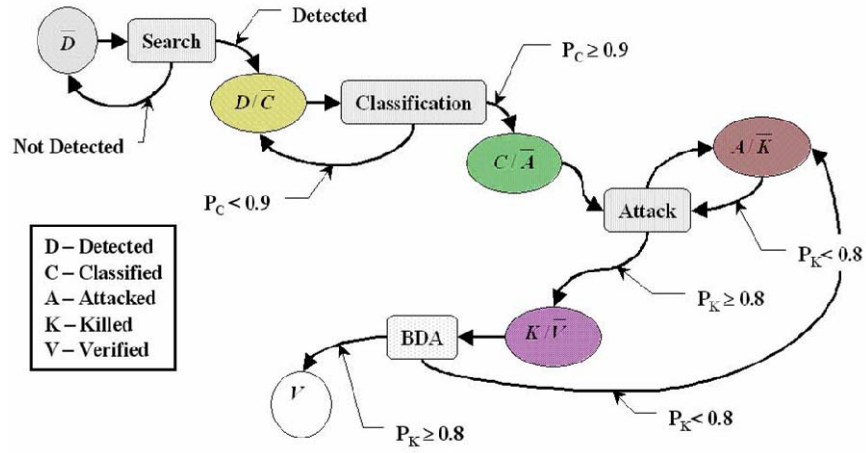
Figure 5. MultiUAV Target State Machine

The possible target states assigned by the simulation to each target are:

- *Target-Not-Detected*

- *Detected-Not-Classified*

- *Classified-Not-Attacked*

- *Attacked-Not-Killed*

- *Killed-Not-Verified*

- *Target-Verified-Killed*

Task assignments are given to each UCAV in the battle space based on an expected state of a target UCAV is engaging. These tasks consist of:

- *Search*

- *Classify*

- *Attack*

- *Verify*

The *Search* task is the nominal state for a UCAV that has not encountered any targets or been assigned to perform a task on a target detected by some other UCAV. In the case of a single UCAV simulation, once a target is detected the task assignment algorithm assumes the kill chain will be completed. All tasks required to complete the kill chain are immediately assigned to the UCAV with an associated completion time. The UCAV completes each task in the appropriate sequence shown in Figure 5 until the target has been verified as killed. In the event a false target is encountered and correctly classified, the *Attack* and *Verify* task assignments are effectively ignored and not completed as assigned. In the case of simulations involving multiple UCAV's, the task assignments are allocated to individual UCAV's based on the specified cooperation algorithm.

Automatic Target Recognition (ATR)

The ATR process is implemented as part of the *Classify* task assignment. When a target enters the sensor footprint of a UCAV for the first time its state is changed from *target-not-detected* to *detected-not-classified*. One or more UCAVs are then assigned the task to classify the target. As shown in Figure 5, the outcome of a classify task is dependent on the parameter P_C. P_C represents the *probability of target identification confidence*. A value for P_C, $0 < P_C < 1$, is given based on the angle of encounter between the path of the UCAV and the orientation of the target as shown in Figure 6 [11].
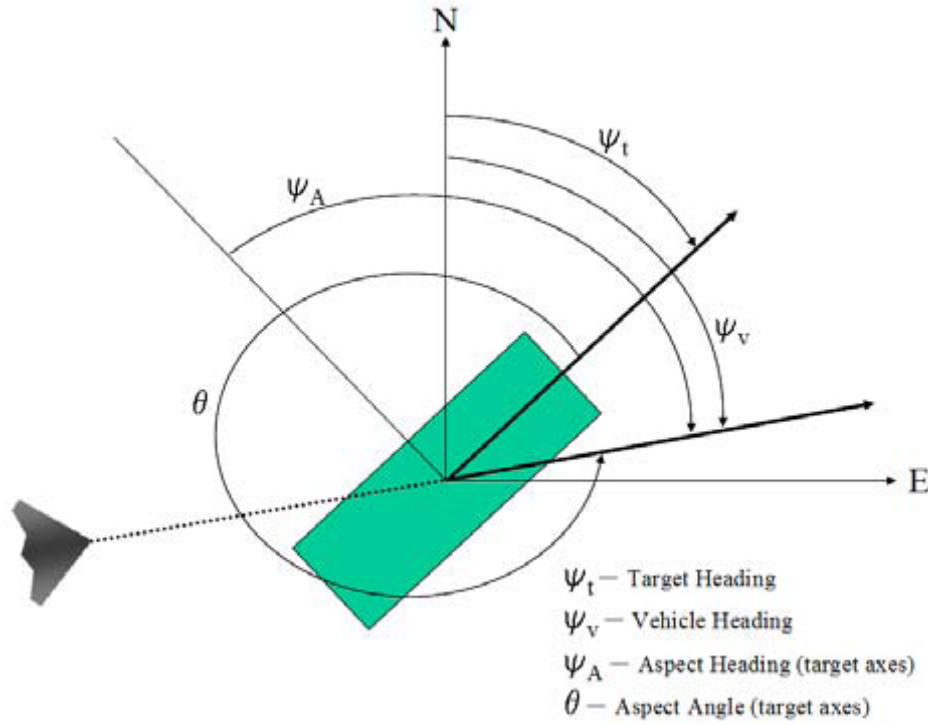
Figure 6. UCAV Angle of Encounter to Target

If the value of P_C meets or exceeds a user-defined *ATR-sensor-threshold* (for example, 0.9 in Figure 5) the target state is changed to *classified-not-attacked*. If the value of P_C is less than the threshold the target state remains *detected-not-classified* and additional classify tasks are executed. The individual values of P_C from each classify task are combined into one value until the threshold criteria is met.

Modifications and Additions to MultiUAV

The following modifications and additions were made to MultiUAV to allow for implementation of the academic scenarios used to accomplish the research objectives.

Target Discovery Times Modification

Although MultiUAV has algorithms to generate a uniform distribution and approximate a Poisson distribution with a fixed quantity of targets, these algorithms were not employed for this research. This was due to a limitation of MultiUAV that only allowed for one target type and one distribution algorithm to be used in generating the target discovery time array. Recall that Scenario 1 required a single target with a uniform distribution and a Poisson field of false targets. To overcome this obstacle the algorithm used to create the target discovery time array was replaced with one that allowed for multiple function calls to any of the target distribution algorithms. The target discovery time array is built by merging the output from each function call into a single array that is then sorted and scaled to the total simulation run time. The actual Matlab *m-file* used is available in Appendix A.

Poisson Distribution Modification

In the current version of MultiUAV, the approximation for a Poisson field of targets contains an exact quantity of targets or false targets equal to the Poisson parameters described by Equations 4 and 6 in Chapter 2. This is inadequate since a Poisson distribution is defined by an expected target density over a defined area or time. It should be expected that two realizations of a Poisson field of targets with identical Poisson parameters will contain different quantities of targets. For this work a new algorithm for generating a Poisson distribution was implemented that built the target discovery time array using a simple Poisson process algorithm. In a Poisson process the intervals between arrival times are exponentially distributed with parameter $1/\lambda$ where λ

32

is the Poisson parameter. The new algorithm utilized the ability of Matlab to generate

random numbers with an exponential distribution to build a Poisson distribution of

targets. The actual simulation subroutine used is reproduced below:

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Subroutine PoissonDiscoveryTimes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [PoissonDiscoveryTimes]=PoissonDistribution(ExpectedTargets)

% This function is called by TargetDiscoveryTimes.m
%
%  Inputs:
%    ExpectedTargets - Number of expected target encounters
%                      of any target type
%
%  Outputs:
%    PoissonDiscoveryTimes - Target discovery times as determined
%                            through a modelled Poisson process
%                            with characterized by ExpectedTargets.
%
%  Note:  Requires Matalb RANDOM toolbox.  If RANDOM toolbox
%         not available use the following line of code:
%         TimeStep(i)=-log(rand)/Lambda;

Lambda=ExpectedTargets; % Poisson parameter
DiscoveryTimes(1)=random('Exponential',1/Lambda); % first tgt arrival time

i=1;
stop=0;
while (DiscoveryTimes(i)<1) && (stop==0),
    TimeStep=random('Exponential',1/Lambda); % next arrival time
    if TimeStep < 1-DiscoveryTimes(i), % check next arrival time < 1
        DiscoveryTimes(i+1)=DiscoveryTimes(i)+TimeStep; % build arrival array
        i=i+1;
    else % stop if next arrival would exceed 1
        stop=1;
    end
end

PoissonDiscoveryTimes=DiscoveryTimes'; % OUTPUT to TargetDiscoveryTimes

Return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Target State Machine Modification

The focus of this research was to quantify the performance of the simulated ATR process in MultiUAV for comparison to analytic models. The data product of interest is the outcome of all simulated ATR classification events. Tracking the state of each target past the *classified-not-attacked* state as described by Figure 5 was not necessary and would have required unnecessary computer processing time. In fact, the previous work of Dunkel and Schulz the number of Monte Carlo simulation runs that could be performed was limited by available computer resources. Schulz was only able to considered 100 Monte Carlo simulations for each scenario design. Fewer simulation runs may introduce unacceptable differences between the mean values produced from the simulation results and the analytic solutions. This is apparent in the results presented in Chapter 4. To allow for more Monte Carlo simulation runs MultiUAV was modified so that once a target was classified the subsequent tasks were never performed. This is the same as how a false target classification is handled. With this modification, significantly more Monte Carlo runs could be performed while requiring significantly less computer processing time.

An additional modification was made to capture the required simulation metrics by altering the target state *classified-not-attack* to included the sub-state *type(n_{truth})-classified-type($n_{declared}$)* with *n* target/declaration types. As previously acknowledged, only one target type and one false target type was considered resulting in the following four possible target states:

- *Target-Classified-Target*

- *Target-Classified-False-Target*

- *False-Target-Classified-False-Target*

- *False-Target-Classified-Target*

This allowed for an easy way to track the simulated ATR classification results. All simulation classification events were recorded in a data file for later comparison to the analytical solutions of Chapter 2.

ATR Confusion Matrix Modification

In the earlier versions of MultiUAV, the probability of correctly identifying an encountered target type was dependent only on the number of *Classify* tasks performed, increasing the value of P_C until the threshold criteria was met. A correct target classification was guaranteed provided the simulation time did not run out before the value of P_C exceeded the threshold value. This does not accurately duplicate the performance of a realistic ATR system.

MultiUAV was modified by Dunkel [3] to incorporate a confusion matrix into the ATR decision model to allow incorrect target declarations, as described in Chapter 2. His modification allowed the user to implement the *Classify* task with either the P_C values as described above or his confusion matrix model for ATR performance. The confusion matrix used in MultiUAV is similar to the one in Table 1 only expanded to allow for some number of *n* target types as shown in Table 2 ($n = 5$). Any target type could be implemented as a false target. Note that the column for each target type must sum to 1.

	Target Type Encountered				
	Type 1	**Type 2**	**Type 3**	**Type 4**	**Type 5**
Type 1	$P_{TR\ 1|T1}$	$P_{TR\ 1|T2}$	$P_{TR\ 1|T3}$	$P_{TR\ 1|T4}$	$P_{TR\ 1|T5}$
Type 2	$P_{TR\ 2|T1}$	$P_{TR\ 2|T2}$	$P_{TR\ 2|T3}$	$P_{TR\ 2|T4}$	$P_{TR\ 2|T5}$
Type 3	$P_{TR\ 3|T1}$	$P_{TR\ 3|T2}$	$P_{TR\ 3|T3}$	$P_{TR\ 3|T4}$	$P_{TR\ 3|T5}$
Type 4	$P_{TR\ 4|T1}$	$P_{TR\ 4|T2}$	$P_{TR\ 4|T3}$	$P_{TR\ 4|T4}$	$P_{TR\ 4|T5}$
Type 5	$P_{TR\ 5|T1}$	$P_{TR\ 5|T2}$	$P_{TR\ 5|T3}$	$P_{TR\ 5|T4}$	$P_{TR\ 5|T5}$

Target Reported As (row axis label)

Table 2. MultiUAV Confusion Matrix

For this research only two target types, a generic target and false target, were considered. The MultiUAV confusion matrix subroutine is shown in Appendix B.

ROC Curve Modification

An additional modification made to the MultiUAV was to mathematically link the values of P_{TR} and P_{FTR} in terms of the ROC curve equations presented in Chapter 2. The modification allows the user to specify a value for P_{TR} that is then used to automatically populate MultiUAV's confusion matrix used to simulate the ATR process for each target encounter. The input value of P_{TR} can be based on empirical data from experimentation with a real sensor/ATR system or given as a purely academic parameter. For this research P_{TR} was given as an academic parameter. P_{FTR} is calculated using one of two equations. Equation 32 is used for simulations of Scenario 2 that were compared to Decker's analytic expressions. Equation 31 is used for simulations of Scenario 1 that

allowed time varying parameters. Equation 31 can be algebraically rearranged to make P_{FTR} an explicit function of P_{TR}.

$$P_{FTR} = 1 - \frac{P_{TR}}{(1-c)P_{TR} + c} \qquad (36)$$

Time Varying ATR Parameters Addition

The benefit of using the ROC curve equation, Equation 36, to define the confusion matrix parameters in MultiUAV is that it provides the vehicle to implement the control input parameters for the dynamic optimization problem formulation described in Chapter 2. These parameters are the area coverage rate, which determines c, and the sensor threshold, which determines P_{TR}. P_{TR} and c can easily be redefined as the time varying inputs $P_{TR}(t)$ and $c(t)$, where t is some target discovery time $0 < t < T_S$. P_{FTR} can then be expressed as a function $P_{FTR}\{P_{TR}(t), c(t)\}$. The input c was held constant for this work.

$$P_{FTR}\{P_{TR}(t), c(t)\} = 1 - \frac{P_{TR}(t)}{(1-c(t))P_{TR}(t) + c(t)} \qquad (37)$$

The value for $P_{TR}(t)$ at a target discovery time t can be input into MultiUAV from a lookup table constructed from Rosario's [12] Matlab numerical dynamic optimization algorithm. For each target discovery time the value for $P_{TR}(t)$ is pulled from the table and input directly into the confusion matrix before a target classification is made. An alternate method not applied in this research is to represent $P_{TR}(t)$ and/or $c(t)$ as closed form equations in the simulation. The simulation subroutines used to model the ROC

curve equations and time varying control inputs as well as build the confusion matrix is
reproduced below:

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Implements Static Ptr using Decker's ROC curve equation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ptr = 0.9;          % INPUT - Static Ptr
q = 18;             % INPUT - ROC curve equation parameter
Pftr = 1 - Ptr^q;   % Decker's ROC curve equation for Pftr

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Implements dynamic Ptr Schedule with Kish's control problem
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load('PtrSchedule.mat');  % Load data file with Ptr schedule
c = 100;            % INPUT - ROC curve equation parameter
lambda = 20;        % INPUT - Poisson parameter for false target density
DiscoveryTime=(TargetDiscoveryTimes(Target.ID);
Ptr = PtrSchedule(DiscoveryTime);   % INPUT - Dynamic control input Ptr
Pftr = 1-Ptr/((1-c)*Ptr+c);         % Kish's ROC curve equation for Pftr

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Build Confusion matrix for True/False targets
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ProbabilityID(1,1)=Ptr;     % Prob. of target report given target
ProbabilityID(2,1)=1-Ptr;   % Prob. of false target report given target
ProbabilityID(1,2)=1-Pftr;  % Prob. of false target report given false target
ProbabilityID(2,2)=Pftr;    % Prob. of target report given false target

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Chapter Summary

The purpose of this chapter is to provide an overview of the MultiUAV
simulation environment and how the academic scenarios developed in Chapter 2 were
incorporated to accomplish the research objectives. The key simulation functions
pertaining to modeling the ATR process and modifications made to MultiUAV were
discussed.

Chapter 4 will present the reduced results from various MultiUAV simulations of the academic scenarios and how they compare to the analytic expressions from Chapter 2. In addition, Chapter 4 will provide a limited analysis on how results from these types of simulations can be used to understand design considerations in the development of wide area search and engagement UCAV systems.

IV. Results and Analysis

Chapter Overview

This Chapter presents the reduced results from various MultiUAV simulations and how they compare to values from the analytic expressions of Chapter 2. The parameters for each simulation scenario were developed with the analytic background theory explained in Chapter 2 and employed using the capabilities and modifications of the MultiUAV simulation as described in Chapter 3. Extensive post simulation data processing was required to convert simulation results into concise graphical and tabular forms.

Scenarios Considered:

Scenario 2 was used to validate the ability of MultiUAV to simulate the performance of an ATR process with a multiple warhead capable UCAV for use in wide area search and destroy research applications (Research Objective 1). Scenario 1 was used to evaluate the ability of MultiUAV to incorporate time varying parameters to dynamically optimize the simulated ATR performance for wide area search and engagement research applications (Research Objective 2).

Simulation Validation for a Multiple Warhead Capable UCAV

The reduced data presented in this section from the simulations of Scenario 2 have been formatted to mimic potential trade studies that may be performed in the design

and operational application of an autonomous wide area search and engagement UCAV system. These trade studies consist of the following cases:

- Case 1: Expected number of attacks vs. probability of target report.

- Case 2: Expected number of attacks vs. warhead quantity.

- Case 3: Probability of specified number of attacks.

Decker [1] suggested that these trade studies can be used to make design decisions in the development of wide area search and engagement UCAV systems as well as determine the resources required for specific operational missions. These results are used to validate the ATR model performance of MultiUAV.

Simulation Input Parameters

The following input parameters were incorporated into MultiUAV to simulate the various studied conditions of Scenarios 2:

- Probability of target report (constant) P_{TR}.

- Sensor threshold parameter q from Equation 32 to calculate P_{FTR}

- Poisson parameter λ_T for target density in A_S.

- Poisson parameter λ_{FTA} for false target density in A_S.

- UCAV warhead quantity, w.

- Number of Monte Carlo simulation runs.

Performance Metrics

For Scenario 2 the reduced simulation data were compared to the following four analytically calculated metrics given by Equations 27, 28, 29, and 30:

- Probability of a specified number of target attacks, $P_{TA}(t=N)$.

- Expected number of target attacks, E_{TA}.

- Probability of a specified number of false target attacks, $P_{FTA}(f=M)$.

- Expected number of false target attacks, E_{FTA}.

Case 1: Expected Number of Attacks vs. Probability of Target Report

Case 1 examines the ability of MultiUAV to accurately simulate the ATR process

performance of a wide area search and engagement system for different given values of

P_{TR}. By evaluating E_{TA} and E_{FTA} as a function of P_{TR} designers may determine the

performance level required to accomplish a given mission. This type of trade study may

produce potential cost savings if it can be determined a less sophisticated and assumedly

cheaper sensor/ATR system can satisfy the mission's performance requirements.

For this study, two sensor/ATR system types were considered with sensor

threshold values of *q=18* and *q=10* from Equation 32. In this case a higher value of *q*

represents a better sensor/ATR system while the lower value represents a less capable but

cheaper system. The distribution of targets and false targets in the battle space are

defined by the Poisson parameters $\lambda_T=10$ and $\lambda_{FT}=20$. The simulated UCAV was armed

with a warhead quantity of *w=10*. The values used for P_{TR} varied from 0.1 to 1 in

increments of 0.1. Because of computer processing time limitations the results presented

here were reduced from 100 Monte Carlo simulation runs for each value of P_{TR}.

The anticipated performance outcome, as shown by Decker [1], should reflect that

for low values of P_{TR} the value of E_{TA} for both sensor/ATR systems will be nearly

identical. This is because for low values of P_{TR} the resulting occurrence of false positive

errors is negligible as determined by the extremely small values for $(1-P_{FTR})$ calculated through the ROC curve equation, Equation 32. As P_{TR} is increased, the value of E_{TA} for both sensor/ATR systems will hit a peak and start to decline as false positive errors becomes more of a factor with increasing values of $(1-P_{FTR})$ (i.e. warheads start to be used on false targets). It is expected the cheaper sensor/ATR system will hit its peak E_{TA} before the better system. As the value of P_{TR} goes to 1 for both systems false positive errors as indicated by E_{FTA} will overwhelm E_{TA} because of the proportionally higher density of false targets in the battle space. This behavior pattern is evident in Figures 7 and 8 below that plot the simulation results and analytic solutions of E_{TA} and E_{FTA} as a function of P_{TR} for the two sensor/ATR systems. Tables 3 and 4 quantify the same results.
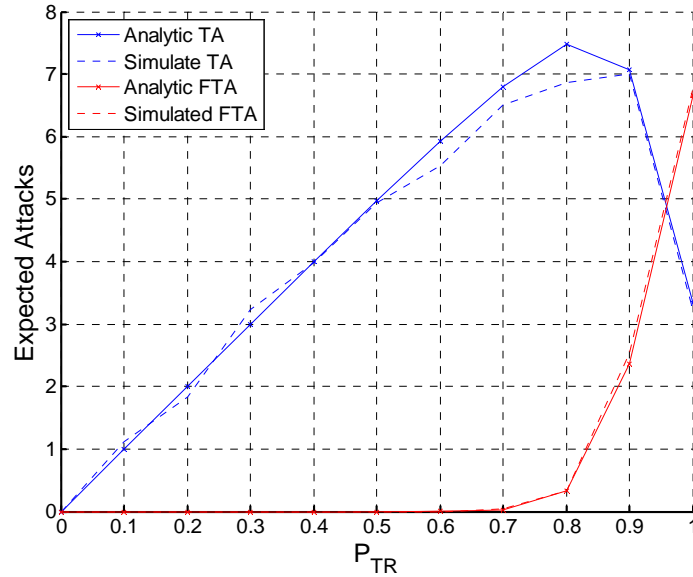
Figure 7. E_{TA} **&** E_{FTA} **vs.** P_{TR} **given** $\lambda_T=10, \lambda_{FT}=20, w=10, q=18$
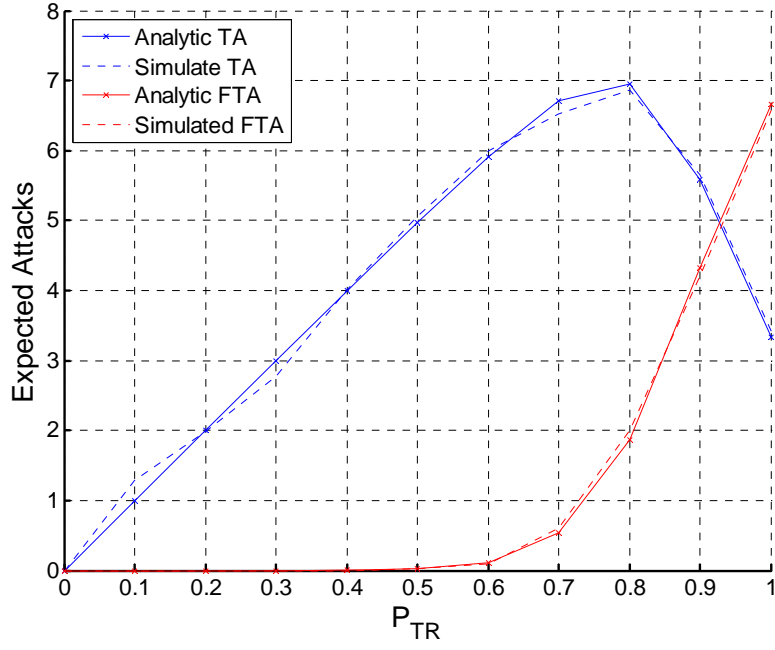
Figure 8. E_{TA} & E_{FTA} vs. P_{TR} given $\lambda_T=10$, $\lambda_{FT}=20$, $w=10$, $q=10$

		E_{TA}			E_{FTA}		
P_{TR}	P_{FTR}	Analytic Value	Simulated Mean	% Diff.	Analytic Value	Simulated Mean	% Diff.
0.10	1.00	1.000	1.110	0.110	0.000	0.000	0.000
0.20	1.00	2.000	1.840	0.080	0.000	0.000	0.000
0.30	1.00	3.000	3.240	0.080	0.000	0.000	0.000
0.40	1.00	3.996	3.980	0.004	0.000	0.000	0.000
0.50	1.00	4.978	4.930	0.010	0.000	0.000	0.000
0.60	0.9999	5.923	5.540	0.065	0.002	0.000	0.000*
0.70	0.998	6.794	6.510	0.042	0.032	0.040	0.250
0.80	0.98	7.487	6.870	0.082	0.337	0.340	0.009
0.90	0.85	7.077	7.000	0.011	2.360	2.540	0.076
1.00	0.00	3.333	3.220	0.034	6.667	6.780	0.017
			Average % Diff.	**0.052**		Average % Diff.	**0.035**

* Because E_{FTA} is extremely small the % diff. is considered equal to 0.000

Table 3. E_{TA} & E_{FTA} vs. P_{TR} given $\lambda_T=10$, $\lambda_{FT}=20$, $w=10$, $q=18$

		E_{TA}			E_{FTA}		
P_{TR}	P_{FTR}	Analytic Value	Simulated Mean	% Diff.	Analytic Value	Simulated Mean	% Diff.
0.10	1.00	1.000	1.290	0.290	0.000	0.000	0.000
0.20	1.00	2.000	1.990	0.005	0.000	0.000	0.000
0.30	1.00	3.000	2.780	0.073	0.000	0.000	0.000
0.40	0.9999	3.996	4.010	0.004	0.002	0.000	0.000*
0.50	0.999	4.977	5.060	0.017	0.019	0.020	0.053
0.60	0.99	5.914	5.990	0.013	0.119	0.090	0.244
0.70	0.97	6.709	6.520	0.028	0.541	0.600	0.109
0.80	0.89	6.950	6.870	0.012	1.866	2.000	0.072
0.90	0.65	5.588	5.650	0.011	4.330	4.200	0.030
1.00	0.00	3.333	3.420	0.026	6.667	6.580	0.013
			Average % Diff.	0.048		Average % Diff.	0.052

* Because E_{FTA} is extremely small the % diff. is considered equal to 0.000

Table 4. E_{TA} & E_{FTA} vs. P_{TR} given λ_T=10, λ_{FT}=20, w=10, q=10

In both Figure 7 and 8 the MultiUAV simulation results, represented by the dashed curves, closely follow the analytic solutions represented by the solid curves. The analytic solution falls within a 99% confidence interval of the simulation results for all simulation runs. In Tables 3 and 4 the largest percent difference between the simulated and analytic solutions is 29% for E_{TA} with P_{TR}=0.1 and q=10. It is important to comment here on the effect of a low number of Monte Carlo simulation runs has on the consistency of the simulated data. In light of the data presented in the following sections for Cases 2 and 3 the larger percent difference values shown here can be said to be entirely due to the relatively low number of Monte Carlo runs.

A potential trade study of the results above may recommend the cheaper sensor/ATR system if low values of P_{TR} are acceptable. For example, with a P_{TR} value of 0.5 (i.e. only half of encountered targets are correctly declared) the simulation generated E_{TA} values of 4.930 for the supposed better sensor/ATR system and 5.060 for the cheaper

system. Note the analytic values of E_{TA} for the two systems are nearly identical at 4.978

and 4.977. In this case E_{FTA} is negligible for both systems. For all practical purposes the

two systems perform at the same level, therefore, the cheaper system should be chosen if

cost is a concern. In contrast, for a P_{TR} of 0.80 or greater the values of E_{TA} are marginally

different while the values for E_{FTA} are at least 0.340 for the better sensor/ATR system and

2.000 for the cheaper system. For higher P_{TR}, the better sensor system should be chosen.

Case 2: Expected Number of Attacks vs. Warhead Quantity

Case 2 examines the ability of MultiUAV to simulate the expected combat

performance of a wide area search and engagement system for different warhead

quantities. By evaluating E_{TA} and E_{FTA} as a function of warhead quantity designers can

better understand vehicle size requirements to accomplish a given mission. Obviously,

more warheads equate to a larger and more expensive UCAV. In addition, this type of

study can be used to determine the number of warheads and/or UCAV's required to

accomplish a specific operational mission.

Two sensor types are examined here with the same sensor threshold values *q=18*

and *q=10* as in Case 1. The distribution of targets and false targets in the battle space is

also the same with the Poisson parameters $\lambda_T=10$ and $\lambda_{FT}=20$. In addition, a P_{TR} value of

0.90 was used. Results are reduced from 1000 Monte Carlo simulation runs for each

sensor type.

The anticipated performance outcome is that the better sensor/ATR system should

be able to achieve higher E_{TA} values and lower E_{FTA} with fewer warheads than the

cheaper system. Figures 9 and 10 below plot the simulation and analytic results for E_{TA}

and E_{FTA} as a function of w for the two sensor/ATR systems modeled. Tables 5 and 6 quantify the same results.
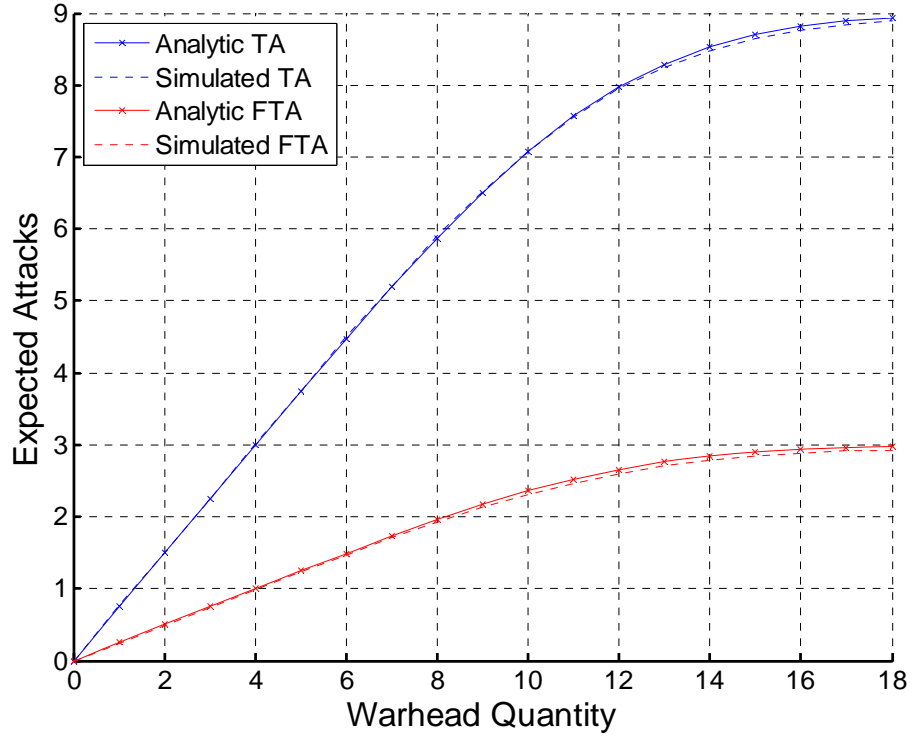
Figure 9. E_{TA} **&** E_{FTA} **vs.** w **given** λ_T=10, λ_{FT}=20, P_{TR}=0.90, q=18

Figure 10. *E_{TA} & E_{FTA} vs. w given λ_T=10, λ_{FT}=20, P_{TR}=0.90, q=10*

	E_{TA}			E_{FTA}		
Warhead Qty	Analytic Value	Simulated Mean	% Diff.	Analytic Value	Simulated Mean	% Diff.
2	1.500	1.511	0.007	0.500	0.489	0.022
4	2.997	3.010	0.004	1.000	0.988	0.012
6	4.476	4.503	0.006	1.493	1.466	0.018
8	5.875	5.906	0.005	1.959	1.917	0.021
10	7.076	7.068	0.001	2.360	2.315	0.019
12	7.970	7.947	0.003	2.658	2.593	0.024
14	8.527	8.474	0.006	2.844	2.791	0.019
16	8.815	8.769	0.005	2.940	2.883	0.019
18	8.938	8.891	0.005	2.981	2.927	0.018
Average % Diff.			**0.005**	Average % Diff.		**0.019**

Table 5. *E_{TA} & E_{FTA} vs. w given λ_T=10, λ_{FT}=20, P_{TR}=0.90, q=18*

Warhead Qty	E_{TA}			E_{FTA}		
	Analytic Value	Simulated Mean	% Diff.	Analytic Value	Simulated Mean	% Diff.
2	1.127	1.150	0.020	0.873	0.850	0.026
4	2.254	2.283	0.013	1.746	1.717	0.017
6	3.379	3.377	0.001	2.619	2.622	0.001
8	4.498	4.549	0.011	3.486	3.439	0.013
10	5.588	5.650	0.011	4.330	4.271	0.014
12	6.598	6.594	0.001	5.113	5.095	0.004
14	7.459	7.415	0.006	5.780	5.780	0.000
16	8.113	8.073	0.005	6.287	6.279	0.001
18	8.547	8.532	0.002	6.623	6.615	0.001
Average % Diff.			0.008	Average % Diff.		0.009

Table 6. E_{TA} & E_{FTA} vs. w given $\lambda_T=10$, $\lambda_{FT}=20$, $P_{TR}=0.90$, $q=10$

The MultiUAV simulation results for E_{TA} and E_{FTA} of both sensor/ATR systems are nearly indistinguishable from the analytic solutions as shown in Figures 9 and 10. The analytic solutions fall within a 99% confidence interval of the simulation results. This strong correlation can be attributed to the relatively high number of Monte Carlo runs as compared to the results from Case 1. Here, the largest percent difference between the simulated and analytic solutions is 2.6% for E_{FTA} with $q=10$ and a warhead quantity of $w=2$. This is negligible when considering it represents a 2.6% difference in a measure of an expected number of attacks.

In a trade study the better sensor/ATR system appears to significantly outperform the cheaper system. The difference in values for E_{FTA} between the two systems is particularly dramatic. As an example, a weapon load of eight warheads produces an E_{FTA} of 1.917 for the better sensor/ATR system and 3.439 for the cheaper system. For the same load the values of E_{TA} is 5.906 for the better system and 4.549 attacks for the cheaper one. Overall, it is apparent that the better sensor/ATR system should be chosen

if there is a specific need to constrain E_{FTA} or limit UCAV size as a function of warhead quantity.

Case 3: Probability of a Specified Number of Attacks

Case 3 examines the ability of MultiUAV to provide data to estimate the probability of at lease some specified number of attacks occurring for a given mission scenario. As in Case 1, this evaluation may give insight into the level of ATR performance required to accomplish a given mission and produce potential cost savings if a cheaper sensor/ATR system can meet the performance requirements. In addition, this type of study may be used to determine the number of warheads or UCAV's required to accomplish a specific operational mission.

The same two sensor types are examined here with the sensor threshold values of $q=18$ and $q=10$. The distribution of targets and false targets in the battle space is again defined by the Poisson parameters $\lambda_T=10$ and $\lambda_{FT}=20$. The warhead quantity specified is $w=10$. The data set is generated from 1000 Monte Carlo simulation runs for each sensor type.

The anticipated performance is that the better sensor/ATR system will maintain higher values of P_{TA} as the specified number of target attacks increases and lower values of P_{FTA} as the number of false target attacks increases. Figures 11 and 12 below plot the simulated and analytic results for P_{TA} and P_{FTA} as a function of at least N target or M false target attacks occurring for the two sensor/ATR systems modeled. Tables 7, 8, 9 and 10 quantify the same results.

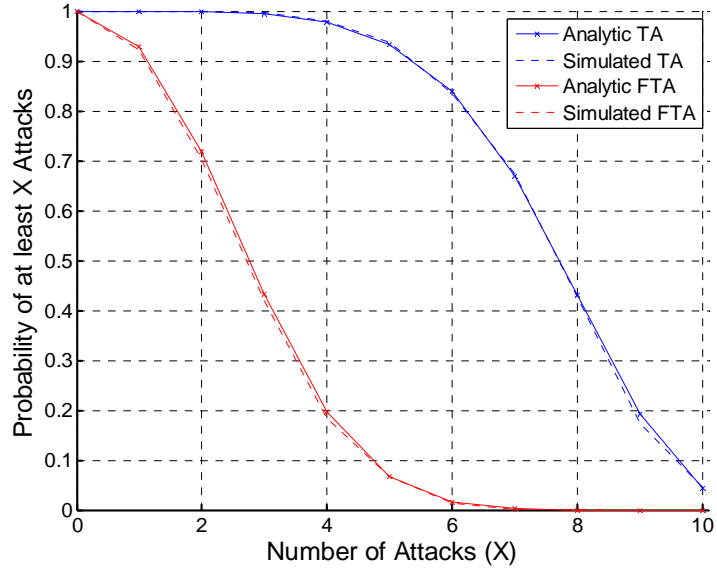**Figure 11.** $P_{Attack}\{TA \text{ or } FTA \geq X\}$ **given** $\lambda_T=10$, $\lambda_{FT}=20$, $P_{TR}=0.90$, $w=10$, $q=18$
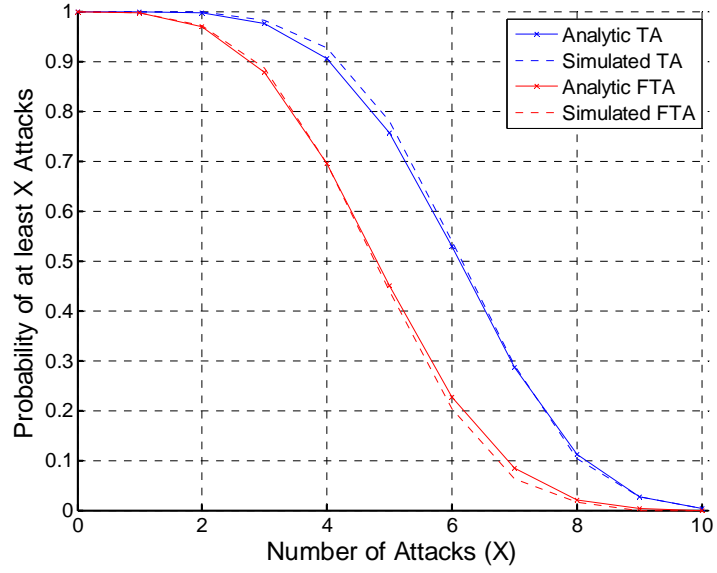

Figure 12. $P_{Attack}\{TA \text{ or } FTA \geq X\}$ **given** $\lambda_T=10$, $\lambda_{FT}=20$, $P_{TR}=0.90$, $w=10$, $q=10$

Possible $N=TA$ Values	Frequency of $N=TA$	$P_{TA}\{TA=N\}$			$P_{TA}\{TA \geq N\}$		
		Analytic	Simulated	% Diff.	Analytic	Simulated	% Diff.
0	0	0.000	0.000	0.000	1.000	1.000	0.000
1	0	0.001	0.000	0.001	1.000	1.000	0.000
2	4	0.005	0.004	0.001	0.999	1.000	0.001
3	17	0.017	0.017	0.000	0.994	0.996	0.002
4	41	0.043	0.041	0.002	0.977	0.979	0.002
5	102	0.094	0.102	0.008	0.934	0.938	0.004
6	162	0.170	0.162	0.008	0.840	0.836	0.004
7	248	0.239	0.248	0.009	0.670	0.674	0.004
8	253	0.239	0.253	0.014	0.431	0.426	0.005
9	127	0.149	0.127	0.022	0.192	0.173	0.019
10	46	0.043	0.046	0.003	0.043	0.046	0.003
Average % Diff.				0.006	Average % Diff.		0.004

Table 7. $P_{TA}\{TA \geq N\}$ given $\lambda_T=10$, $\lambda_{FT}=20$, $P_{TR}=0.90$, $w=10$, $q=18$

Possible $M=FTA$ Values	Frequency of $M=FTA$	$P_{TA}\{FTA=M\}$			$P_{TA}\{FTA \geq M\}$		
		Analytic	Simulated	% Diff.	Analytic	Simulated	% Diff.
0	77	0.072	0.077	0.005	1.000	1.000	0.000
1	219	0.210	0.219	0.009	0.928	0.923	0.005
2	283	0.286	0.283	0.003	0.718	0.704	0.014
3	237	0.236	0.237	0.001	0.432	0.421	0.011
4	117	0.130	0.117	0.013	0.196	0.184	0.012
5	54	0.050	0.054	0.004	0.066	0.067	0.001
6	11	0.013	0.011	0.002	0.016	0.013	0.003
7	1	0.002	0.001	0.001	0.003	0.002	0.001
8	1	0.000	0.001	0.001	0.000	0.001	0.001
9	0	0.000	0.000	0.000	0.000	0.000	0.000
10	0	0.000	0.000	0.000	0.000	0.000	0.000
Average % Diff.				0.004	Average % Diff.		0.004

Table 8. $P_{FTA}\{FTA \geq M\}$ given $\lambda_T=10$, $\lambda_{FT}=20$, $P_{TR}=0.90$, $w=10$, $q=18$

Possible N=TA Values	Frequency of N=TA	$P_{TA}\{TA=N\}$			$P_{TA}\{TA \geq N\}$		
		Analytic	Simulated	% Diff.	Analytic	Simulated	% Diff.
0	0	0.000	0.000	0.000	1.000	1.000	0.000
1	0	0.004	0.000	0.004	1.000	1.000	0.000
2	19	0.021	0.019	0.002	0.996	1.000	0.004
3	54	0.069	0.054	0.015	0.975	0.981	0.006
4	148	0.150	0.148	0.002	0.906	0.927	0.021
5	240	0.228	0.240	0.012	0.756	0.779	0.023
6	249	0.241	0.249	0.008	0.528	0.539	0.011
7	186	0.176	0.186	0.010	0.287	0.290	0.003
8	78	0.084	0.078	0.006	0.111	0.104	0.007
9	22	0.024	0.022	0.002	0.027	0.026	0.001
10	4	0.003	0.004	0.001	0.003	0.004	0.001
Average % Diff.				0.006	**Average % Diff.**		0.007

Table 9. $P_{TA}\{TA \geq N\}$ given λ_T=10, λ_{FT}=20, P_{TR}=0.90, w=10, q=10

Possible M=FTA Values	Frequency of M=FTA	$P_{TA}\{FTA=M\}$			$P_{TA}\{FTA \geq M\}$		
		Analytic	Simulated	% Diff.	Analytic	Simulated	% Diff.
0	4	0.004	0.004	0.000	1.000	1.000	0.000
1	25	0.027	0.025	0.002	0.996	0.996	0.000
2	84	0.091	0.084	0.007	0.969	0.971	0.002
3	193	0.183	0.193	0.010	0.878	0.887	0.009
4	254	0.244	0.254	0.010	0.695	0.694	0.001
5	236	0.224	0.236	0.012	0.451	0.440	0.011
6	142	0.143	0.142	0.001	0.227	0.204	0.023
7	45	0.063	0.045	0.018	0.084	0.062	0.022
8	17	0.018	0.017	0.001	0.021	0.017	0.004
9	0	0.003	0.000	0.003	0.003	0.000	0.003
10	0	0.000	0.000	0.000	0.000	0.000	0.000
Average % Diff.				0.006	**Average % Diff.**		0.007

Table 10. $P_{FTA}\{FTA \geq M\}$ given λ_T=10, λ_{FT}=20, P_{TR}=0.90, w=10, q=10

Again, the MultiUAV simulation results for both sensor/ATR systems are nearly indistinguishable from the analytic solutions as shown in Figures 11 and 12. The analytic

solutions fall within a 99% confidence interval of the simulation results. In this case the largest percent difference between the simulated and analytic solutions is 2.3%. This is negligible when considering it represents a 2.3% difference in the probability of at least some number of attacks occuring.

The better sensor/ATR system appears to outperform the cheaper system as expected. For example, the probability that the better system will yield at least six target attacks is 0.840 compared to 0.528 for the cheaper one. The probability that the better sensor system will results in at least six false target attacks is 0.016 for the better system and 0.227 for the cheaper one. This type of trade study should conclude the better ATR system is preferred if there is a specific need for higher values of E_{TA} for a mission or there is a requirement to constrain E_{FTA}.

Simulation Evaluation for Time Varying Control Parameters

The reduced data presented in this section are from MultiUAV simulations that incorporated time varying control input schedules for P_{TR}. The time scheduled values for P_{TR} were obtained from Rosario's [12] numeric Matlab dynamic optimization as presented in Chapter 2. The following Scenario 1 variations were considered:

- Case 1a: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 1.0$, $c=100$, $\lambda_{FT}=25$

- Case 1b: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 0.2$, $c=100$, $\lambda_{FT}=25$

- Case 2a: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 1.0$, $c=100$, $\lambda_{FT}=5$

- Case 2b: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 0.2$, $c=100$, $\lambda_{FT}=5$

- Case 3a: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 1.0$, $c=50$, $\lambda_{FT}=25$

- Case 3b: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 0.2$, $c=50$, $\lambda_{FT}=25$

- Case 4a: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 1.0$, $c=50$, $\lambda_{FT}=5$

- Case 4b: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 0.2$, $c=50$, $\lambda_{FT}=5$

For each of the above cases studied a parallel static optimization simulation is presented as a basis for comparison. The static optimization formulation provides a constant value of P_{TR} that is intended to maximize P_{TA} subject to the same constrained values of P_{FTA}. The static values for P_{TR} were also obtained from Rosario's [12] Matlab optimization.

Simulation Input Parameters

The following input parameters were required by MultiUAV to allow dynamic and static optimization simulations of Scenario 1:

- Probability of target report (static and time varying) $P_{TR}(t)$.

- Sensor threshold parameter c from Equation 31 to calculate P_{FTR}

- Poisson parameter λ_{FTA} for false target density in A_S.

- Number of Monte Carlo simulation runs.

The maximum allowed constraint P_{FTA} values were chosen to model an unconstrained optimization case with $P_{FTA}=1.0$ and a highly constrained case with $P_{FTA}=0.20$. The values for the ROC curve equation parameter c in the case studies defined above were chosen to model a high quality sensor/ATR system with $c=100$ and a low quality system with $c=50$. The values of λ_{FT} were chosen to model a high false target

density in the battle space with $\lambda_{FT} = 25$ and a low target density with $\lambda_{FT} = 5$ for each sensor/ATR system.

Performance Metrics

For Scenario 1, the reduced simulation data were compared to the numeric dynamic and static optimization output data from Rosario's [12] Matlab algorithm consisting of:

- Probability of target attack, P_{TA}.

- Probability of false target attack, P_{FTA}.

It is expected that the simulations implementing time varying P_{TR} schedules will produce higher final values of P_{TA} than simulations using static P_{TR} values. It is also expected that the unconstrained simulations (i.e. $P_{FTA} \leq 1.0$) will produce higher final values of both P_{TA} and P_{FTA} than the constrained simulations ($P_{FTA} \leq 0.2$).

The results from each MultiUAV simulation are presented in Figures 13-28 and quantified in Tables 11 and 12 in the following pages.

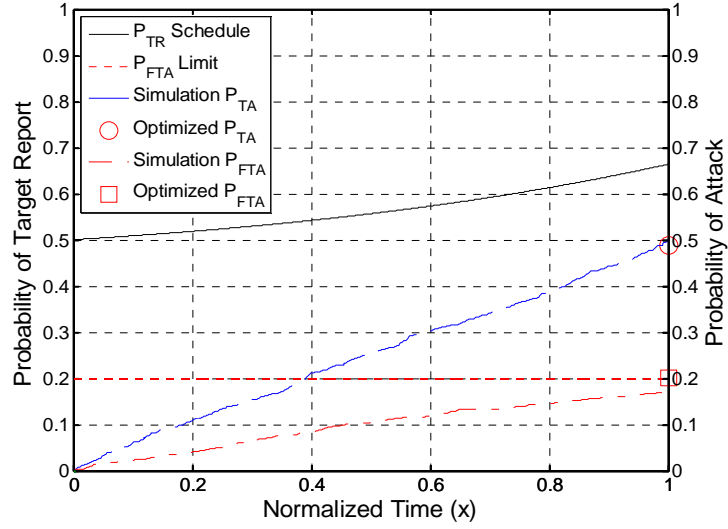*Case 1a: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 1.0$, c=100, λ_{FT}=25*
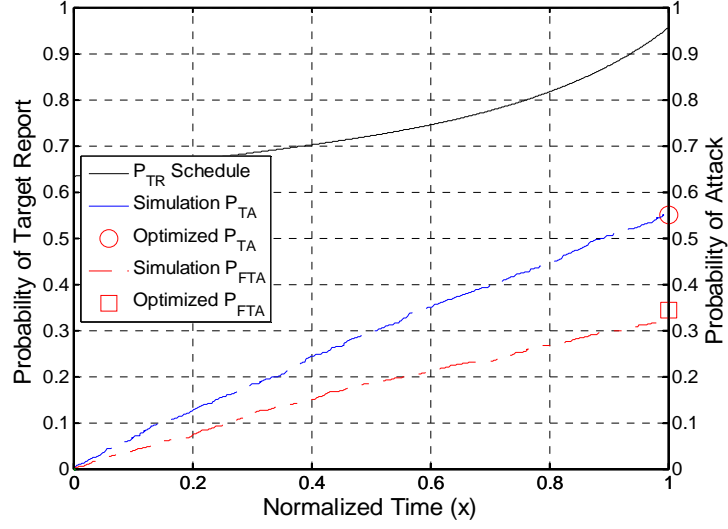
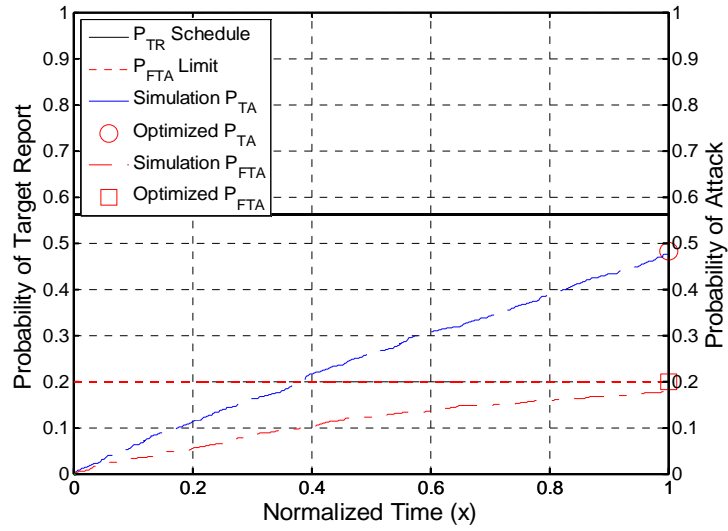**Figure 13. Dynamic Optimization:** *$P_{FTA} \leq 1.0$, c=100, λ_{FT}=25*
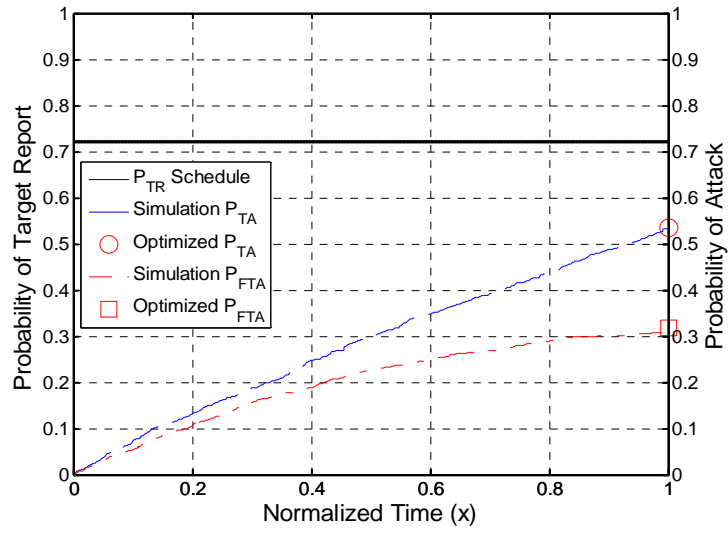
Figure 14. Static Optimization: *$P_{FTA} \leq 1.0$, c=100, λ_{FT}=25*

Case 1b: Maximize: P_{TA}, Subject to: $P_{FTA} \le 0.2$, c=100, λ_{FT}=25

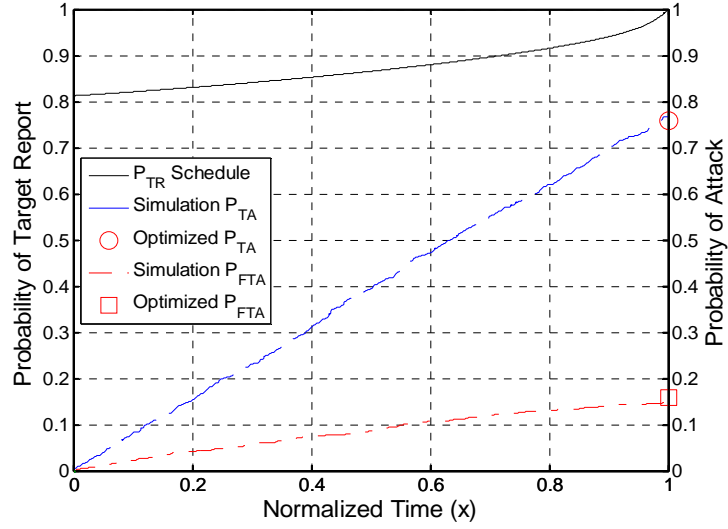
Figure 15. Dynamic Optimization: $P_{FTA} \le 0.2$, c=100, λ_{FT}=25
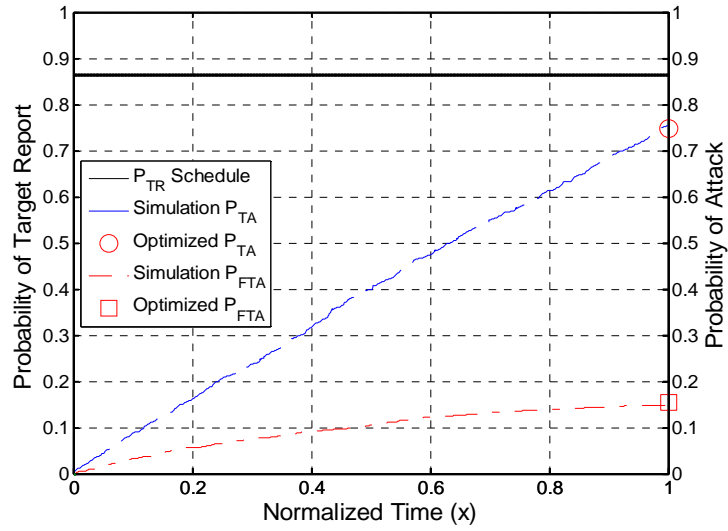

Figure 16. Static Optimization: $P_{FTA} \le 0.2$, c=100, λ_{FT}=25

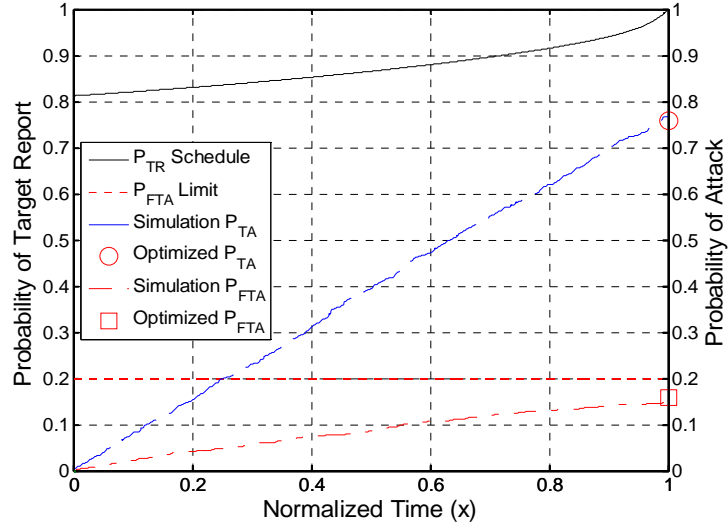*Case 2a: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 1.0$, c=100, λ_{FT}=5*

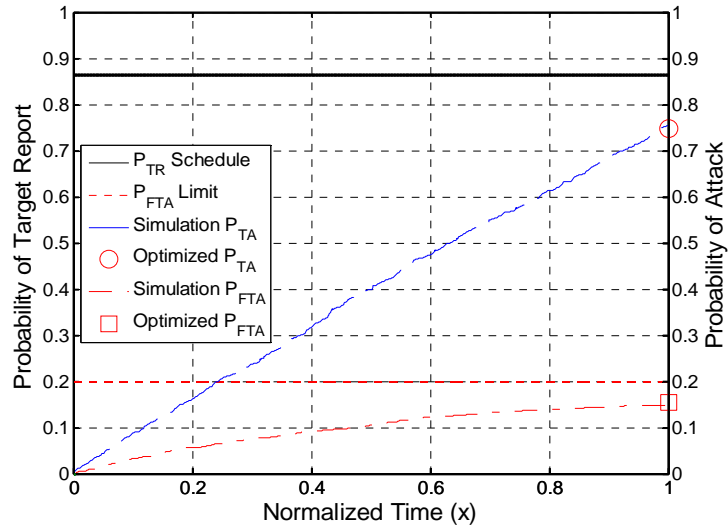**Figure 17. Dynamic Optimization: $P_{FTA} \leq 1.0$, c=100, λ_{FT}=5**

Figure 18. Static Optimization: $P_{FTA} \leq 1.0$, c=100, λ_{FT}=5

Case 2b: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 0.2$, c=100, λ_{FT}=5

Figure 19. Dynamic Optimization: $P_{FTA} \leq 0.2$, c=100, λ_{FT}=5

Figure 20. Static Optimization: $P_{FTA} \leq 0.2$, c=100, λ_{FT}=5

Case 3a: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 1.0$, c=50, λ_{FT}=25

Figure 21. Dynamic Optimization: *$P_{FTA} \leq 1.0$, c=50, λ_{FT}=25*

Figure 22. Static Optimization: *$P_{FTA} \leq 1.0$, c=50, λ_{FT}=25*

Case 3b: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 0.2$, c=50, $\lambda_{FT}=25$

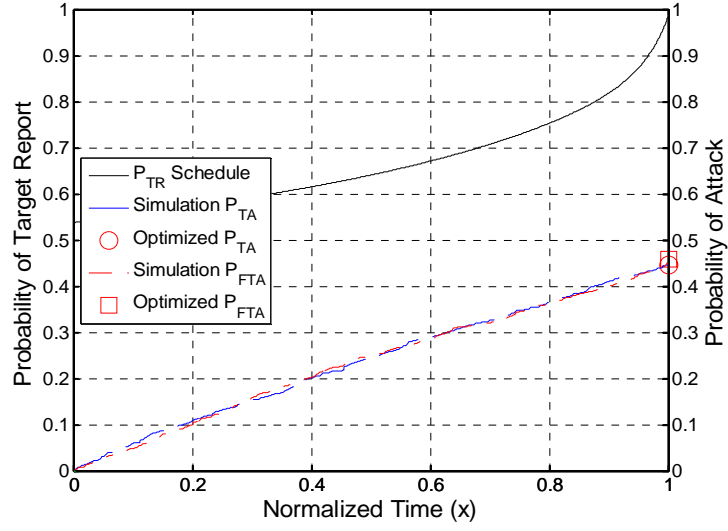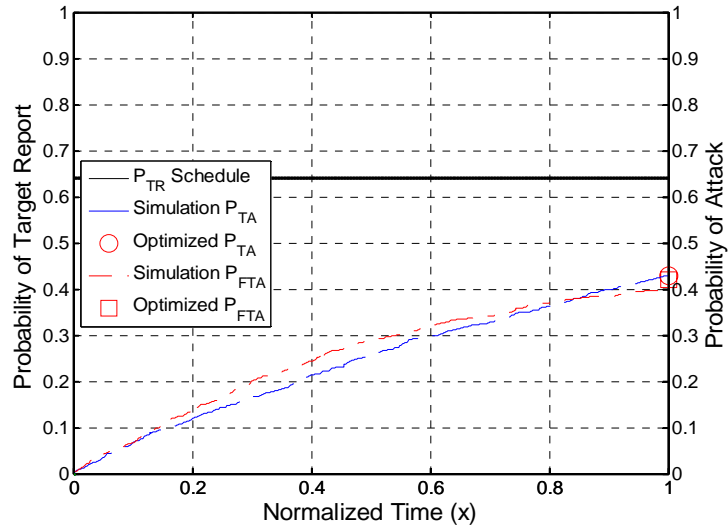**Figure 23. Dynamic Optimization: *$P_{FTA} \leq 0.2$, c=50, $\lambda_{FT}=25$***

Figure 24. Static Optimization: *$P_{FTA} \leq 0.2$, c=50, $\lambda_{FT}=25$*

62

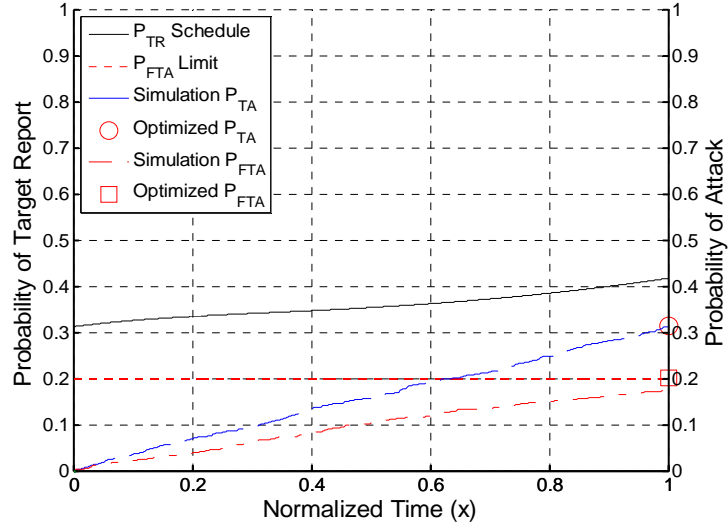*Case 4a: Maximize: P_{TA}, Subject to: $P_{FTA} \le 1.0$, c=50, λ_{FT}=5*

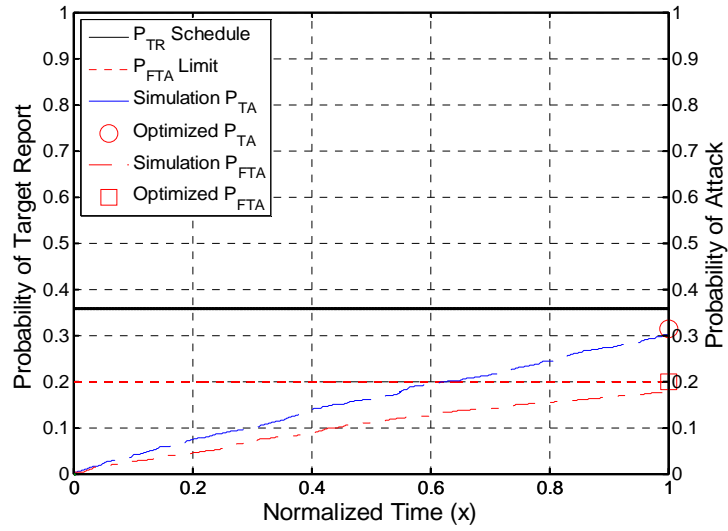**Figure 25. Dynamic Optimization: $P_{FTA} \le 1.0$, c=50, λ_{FT}=5**

Figure 26. Static Optimization: $P_{FTA} \le 1.0$, c=50, λ_{FT}=5

63

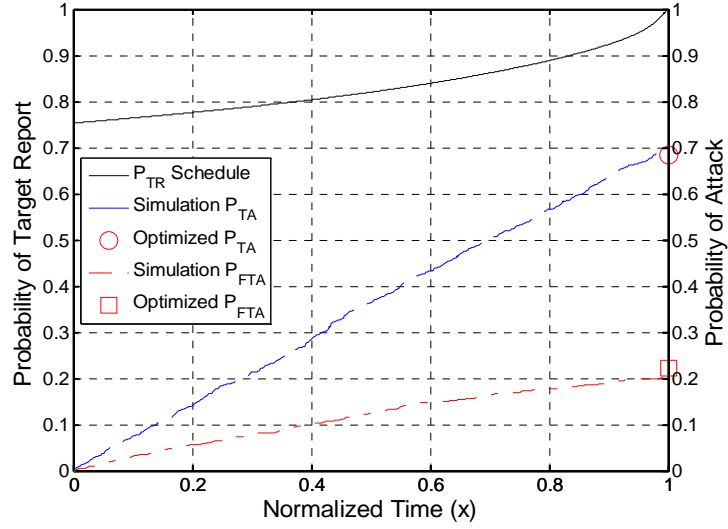*Case 4b: Maximize: P_{TA}, Subject to: $P_{FTA} \leq 0.2$, c=50, λ_{FT}=5*

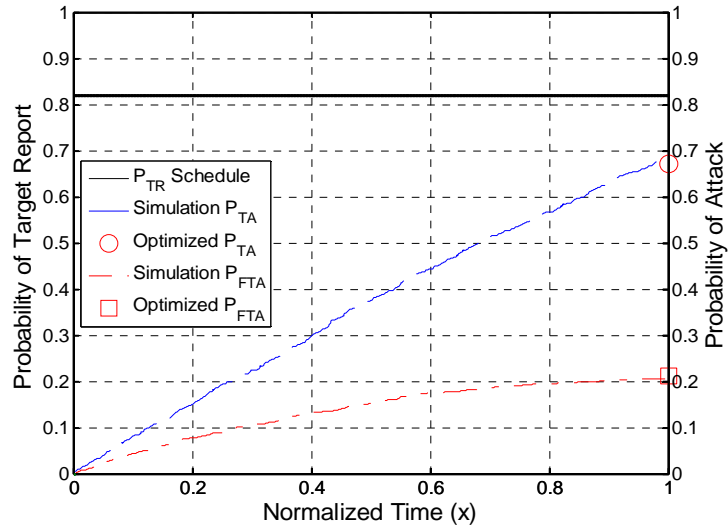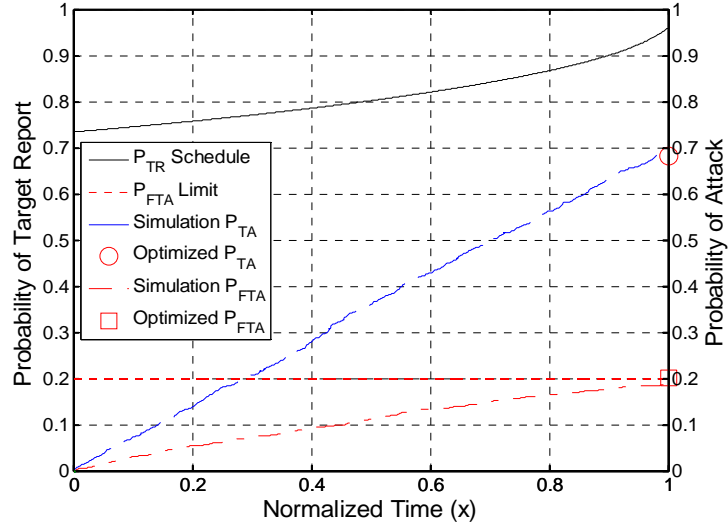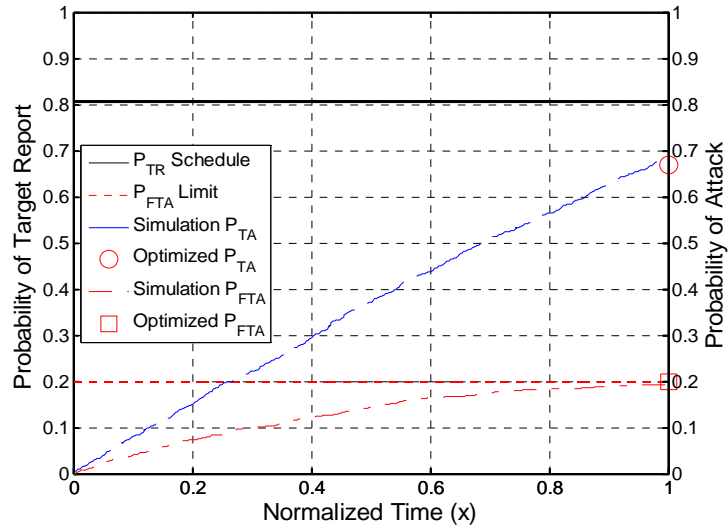**Figure 27. Dynamic Optimization:** *$P_{FTA} \leq 0.2$, c=50, λ_{FT}=5*

Figure 28. Static Optimization: *$P_{FTA} \leq 0.2$, c=50, λ_{FT}=5*

	Dynamic Max P_{TA}			Dynamic Final P_{TFA}		
	Numeric Solution	Simulated Results	% Diff.	Numeric Solution	Simulated Results	% Diff.
Case 1a: $c=100$, $\lambda_{FT}=25$, $P_{FTA} \leq 1.0$	0.551	0.552	0.001	0.343	0.325	0.018
Case 1b: $c=100$, $\lambda_{FT}=25$, $P_{FTA} \leq 0.2$	0.487	0.496	0.009	0.200	0.172	0.028
Case 2a: $c=100$, $\lambda_{FT}=5$, $P_{FTA} \leq 1.0$	0.759	0.768	0.009	0.159	0.148	0.011
Case 2b: $c=100$, $\lambda_{FT}=5$, $P_{FTA} \leq 0.2$	0.759	0.768	0.009	0.159	0.148	0.011
Case 3a: $c=50$, $\lambda_{FT}=25$, $P_{FTA} \leq 1.0$	0.446	0.446	0.000	0.459	0.449	0.010
Case 3b: $c=50$, $\lambda_{FT}=25$, $P_{FTA} \leq 0.2$	0.314	0.311	0.003	0.200	0.176	0.024
Case 4a: $c=50$, $\lambda_{FT}=5$, $P_{FTA} \leq 1.05$	0.685	0.697	0.012	0.223	0.206	0.017
Case 4b: $c=50$, $\lambda_{FT}=5$, $P_{FTA} \leq 0.2$	0.683	0.694	0.011	0.200	0.188	0.012
Average % Diff.			**0.007**	Average % Diff.		**0.016**

Table 11. Dynamic Optimization Results P_{TA} and P_{FTA}

	Static Max P_{TA}			Static Final P_{TFA}		
	Numeric Solution	Simulated Results	% Diff.	Numeric Solution	Simulated Results	% Diff.
Case 1a: $c=100$, $\lambda_{FT}=25$, $P_{FTA} \leq 1.0$	0.535	0.533	0.002	0.318	0.308	0.010
Case 1b: $c=100$, $\lambda_{FT}=25$, $P_{FTA} \leq 0.2$	0.482	0.476	0.006	0.200	0.179	0.021
Case 2a: $c=100$, $\lambda_{FT}=5$, $P_{FTA} \leq 1.0$	0.747	0.754	0.007	0.153	0.148	0.005
Case 2b: $c=100$, $\lambda_{FT}=5$, $P_{FTA} \leq 0.2$	0.747	0.754	0.007	0.153	0.148	0.005
Case 3a: $c=50$, $\lambda_{FT}=25$, $P_{FTA} \leq 1.0$	0.430	0.428	0.002	0.420	0.398	0.022
Case 3b: $c=50$, $\lambda_{FT}=25$, $P_{FTA} \leq 0.2$	0.313	0.301	0.012	0.200	0.179	0.021
Case 4a: $c=50$, $\lambda_{FT}=5$, $P_{FTA} \leq 1.05$	0.670	0.685	0.015	0.212	0.205	0.007
Case 4b: $c=50$, $\lambda_{FT}=5$, $P_{FTA} \leq 0.2$	0.670	0.685	0.015	0.200	0.192	0.008
Average % Diff.			**0.008**	Average % Diff.		**0.012**

Table 12. Static Optimization Results P_{TA} and P_{FTA}

The results presented in the figures and tables above provide strong evidence that MultiUAV is able to incorporate time varying control inputs to conduct dynamic optimization studies of wide area search and engagement research applications. The performance measures produced by Rosario's numeric optimization algorithm, designated as the *Numeric Solution* in the Tables 11 and 12, are within a 99% confidence interval of the simulation results.

As expected, the above results also show that the simulations implementing time varying P_{TR} schedules consistently produced higher final values of P_{TA} than the simulations using static P_{TR} values. In addition, the results show that the unconstrained simulations, where $P_{FTA} \leq 1.0$, produced higher final values of P_{TA} and P_{FTA} than the constrained simulations with the exception of Case 2. In the constrained simulation of Case 2, shown by Figure 19, the final value for P_{FTA} is significantly lower than the max allowed value of 0.2. The dynamic optimization results from Rosario's algorithm show this as well in Tables 11 and 12. This implies that the constraint on P_{FTA} is inactive and therefore has no impact on the resulting schedule for P_{TR}. This allowed for identical results for the constrained and unconstrained simulations.

Summary of Results and Analysis

The results presented in this chapter have demonstrated the ability of MultiUAV to simulate the performance of an ATR process with a multiple warhead capable UCAV for use in wide area search and destroy research applications (Research Objective 1). This was accomplished through the use of several hypothetical trade studies performed using data generated by various MultiUAV simulations and comparing this data to analytic theory. The data presented in this chapter also demonstrated the ability of MultiUAV to simulate time varying control parameters to dynamically optimize ATR performance for wide area search and engagement research applications (Research Objective 2). Chapter 5 will present the final conclusions and recommendations of this work.

V. Conclusions and Recommendations

Conclusions

This work presented a method to validate the ability of a simulation environment such as MultiUAV to realistically model the performance of an ATR process to discriminate between targets and false targets in wide area search and engagement applications. This was accomplished by comparing results generated from multiple simulations of academically contrived wide area search and engagement scenarios to closed form analytic expressions derived for the same scenarios. The ability of MultiUAV to simulate the performance of an ATR process of a multiple warhead capable UCAV and the ability of MultiUAV to incorporate time varying parameters to dynamically optimize ATR performance were examined. Also presented in this research was a limited analysis on how to utilize these types of simulations to make design decisions in the development of wide area search and engagement UCAV systems as well as determine asset allocations (i.e. UCAV quantity and/or warhead payload) for specific operational missions.

Contributions

The primary goal of this research was to develop a methodology to validate the baseline performance of a simulation environment for wide area search and engagement research applications. Extensive modification to the subroutines of MultiUAV was required to fulfill this objective. An additional contribution of this work was the drawing together of the key algorithms required to simulate a realistic ATR process. These

67

algorithms alone can be the basis for new simulation environments and future research applications.

Limitations of Research

The simulations demonstrated here were limited to very simplistic design parameters intended to exactly duplicate academic scenarios. This work does not suggest that MultiUAV is "off the shelf" ready to model an ATR process as presented for a cooperative group of UCAVs. Only a single UCAV was simulated in this work which required extensive modifications to MultiUAV to produce meaningful results. Additional work is required to model the ATR process of multiple UCAV's to cooperatively classifying encountered targets.

Recommendations for Future Research

1. Extend this work with MultiUAV to include the other academic scenarios with a multiple warhead capable UCAV. Continue modifications of MultiUAV to allow for multiple UCAVs to cooperatively classify and engage encountered targets using a modified form of the confusion matrix ATR process presented here. In addition, consider simulation of operational scenarios that do not necessarily lend themselves to easy analytic evaluation.

2. Pursue development of an alternative simulation environment to MultiUAV specifically for AFIT research applications. The modifications made to MultiUAV in this line of research represent the core simulation functions required to model wide area search and engagement vehicles. MultiUAV in its current form employs high fidelity

vehicle dynamic models and complicated communication subroutines that are unnecessary for some research applications. A new simulation could be developed to perform the same simulations as MultiUAV with a simple point mass vehicle models and stochastic subroutines to model ATR, weapon employment, and other simulated functions.

3. Use MultiUAV or an alternative simulation environment to support the parallel development of a small scale experimental wide area search and engagement vehicle.

Summary

The use of computer simulation in the development of cooperatively controlled unmanned combat aerial vehicles (UCAV) for autonomous wide area search and engagement applications was addressed. The objective of this research was to demonstrate a method to validate a simulation's ATR model for future use in the evaluation of cooperative control schemes. This was accomplished by comparing the results from multiple simulations of academically contrived wide area search and engagement scenarios to closed form analytic solutions derived for the same scenarios.

Appendix A: MultiUAV Target Discovery Time Subroutines

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Subroutine TargetDiscoveryTimes.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [TargetDiscoveryTimes] = TargetDiscoveryTimes(SearchTime,...
                                                   Targets,FalseTargets)

% This function is called by TargetSetup.m
%
%  Inputs:
%    SearchTime - Total simulation run time
%    Targets - Expected number of target encounters
%    FalseTargets - Expected number of false target encounters
%
%  Outputs:
%    TargetDiscoveryTimes(:,1) - Target discovery time array used to
%                                determine spatial positions of targets
%                                based on search agent location at
%                                the discovery time.
%    TargetDiscoveryTimes(:,2) - Links target type (i.e. target or false
%                                target) to adjacent discovery time.
%
%  AFIT/ENY
%  December 2006 - Created and Debugged - Marlin
%

if nargin==0,
    SearchTime=1; % INPUT - total search time
    Targets=1;     % INPUT - expected real targets
    FalseTargets=20;   % INPUT - expected false targets
end

% call UniformDistribution
[UniformDiscoveryTimes]=UniformDistribution(Targets)

% associated target type
UniformDiscoveryTimes(:,2)=ones(1,length(UniformDiscoveryTimes));

% call PoissonDistribution
[PoissonDiscoveryTimes]=PoissonDistribution(FalseTargets)

% associated target type
PoissonDiscoveryTimes(:,2)=ones(1,length(PoissonDiscoveryTimes))*2;

% sort TargetDiscoveryTimes in sequential order
TargetDiscoveryTimes=sortrows([UniformDiscoveryTimes,
                               PoissonDiscoveryTimes],1);

% scale TargetDiscoveryTimes to SearchTime
TargetDiscoveryTimes(:,1)=TargetDiscoveryTimes(:,1)*SearchTime;

return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Subroutine PoissonDiscoveryTimes.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [PoissonDiscoveryTimes]=PoissonDistribution(ExpectedTargets)
```

```matlab
% This function is called by TargetDiscoveryTimes.m
%
%  Inputs:
%    ExpectedTargets - Number of expected target encounters
%                      of any target type
%
%  Outputs:
%    PoissonDiscoveryTimes - Target discovery times as determined
%                            through a modeled Poisson process
%                            with characterized by ExpectedTargets.
%
%  Note:  Requires Matalb RANDOM toolbox.  If RANDOM toolbox
%         not available use the following line of code:
%         TimeStep(i)=-log(rand)/Lambda;
%
%  AFIT/ENY
%  December 2006 - Created and Debugged - Marlin
%

Lambda=ExpectedTargets; % INPUT - Poisson parameter
DiscoveryTimes(1)=random('Exponential',1/Lambda); % first tgt arrival time

i=1;stop=0;
while (DiscoveryTimes(i)<1) && (stop==0),
    TimeStep=random('Exponential',1/Lambda);  % next arrival time
    if TimeStep < 1-DiscoveryTimes(i),  % tgt arrival times cannot exceed 1
        DiscoveryTimes(i+1)=DiscoveryTimes(i)+TimeStep;  % build arrivals
        i=i+1;
    else % stop if next arrival would exceed 1
        stop=1;
    end
end

PoissonDiscoveryTimes=DiscoveryTimes';  % OUTPUT to TargetDiscoveryTimes

return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% UniformDiscoveryTimes.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [UniformDiscoveryTimes]=UniformDistribution(NumberTargets)

% This function is called by TargetDiscoveryTimes.m
%
%  Inputs:
%    ExpectedTargets - Number of target encounters specified
%
%  Outputs:
%    UniformDiscoveryTimes - Target discovery times as determined
%                            through a modeled uniform distribution.
%
%  AFIT/ENY
%  December 2006 - Created and Debugged - Marlin
%

UniformDiscoveryTimes=rand(NumberTargets,1); % OUTPUT to TargetDiscoveryTimes

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

return
```

71

Appendix B: MultiUAV Classify Targets Subroutines

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Subroutine ClassifyTargets.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Classification = ClassifyTarget(TrueTargetType,Pid)

% This function is called by ATRFunctions.m
%
% Classification - Determines what the vehicle classifies the target as
%                  based on the vehicle's probability of identification
%                  (Pid) and a random draw.  This is not necessarily what
%                  the target truly is.
%
%  Inputs:
%    TrueTargetType - True type of the target
%    Pid            - Confusion matrix
%
%                        Encountered Object
%                    1     2     3     4     5
%            Pid = [0.000 0.000 0.000 0.000 0.000;...  % 1
%                    0.000 0.000 0.000 0.000 0.000;...  % 2
%                    0.000 0.000 0.000 0.000 0.000;...  % 3  Declared Object
%                    0.000 0.000 0.000 0.000 0.000;...  % 4
%                    0.000 0.000 0.000 0.000 0.000];    % 5
%
%  Outputs:
%    Classification - What the vehicle classified the target as
%
%  AFIT/ENY
%  September 2001 - Created and Debugged - Dunkel

RandomNumber = rand;
j = TrueTargetType;
Bounds = [Pid(1,j); ...
          Pid(1,j)+Pid(2,j); ...
          Pid(1,j)+Pid(2,j)+Pid(3,j); ...
          Pid(1,j)+Pid(2,j)+Pid(3,j)+Pid(4,j); ...
          Pid(1,j)+Pid(2,j)+Pid(3,j)+Pid(4,j)+Pid(5,j)];

% The variable Bounds is used to determine the
% relation between the random number and the
% confusion matrix.  The elements of bounds are
% simply the progressive summation of the elements
% in a given column of the confusion matrix (Pid).

if RandomNumber <= Bounds(1)
   Classification = 1;  % Target State = type(j)-classified-type(1)
elseif RandomNumber > Bounds(1) & RandomNumber <= Bounds(2)
   Classification = 2;  % Target State = type(j)-classified-type(2)
elseif RandomNumber > Bounds(2) & RandomNumber <= Bounds(3)
   Classification = 3;  % Target State = type(j)-classified-type(3)
elseif RandomNumber > Bounds(3) & RandomNumber <= Bounds(4)
   Classification = 4;  % Target State = type(j)-classified-type(4)
elseif RandomNumber > Bounds(4) & RandomNumber <= Bounds(5)
   Classification = 5;  % Target State = type(j)-classified-type(5)
end

return;
```

Bibliography

1. Decker, Douglas D. *Decision factors for Cooperative Multiple Warhead UAV Target Classification and Attack with Control Applications*. PhD Dissertation, AFIT/DS/ENY/05-04, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, October 2004.

2. Decker, Douglas D., David R. Jacques and Meir Pachter. *A Theory of Wide Area Search and Engagement*. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 2006.

3. Dunkel, Robert E. *Investigation of Cooperative Behavior in Autonomous Wide Area Search Munitions*. MS Thesis, AFIT/GAE/ENY/02-4, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2002.

4. Gillen, Daniel P. *Cooperative Behavior Schemes for Improving the Effectiveness of Autonomous Wide Area Search Munitions*. MS Thesis, AFIT/GAE/ENY/01M-03, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2001.

5. Gozaydin, Orhan. *Analysis of Cooperative Behavior for Autonomous Wide Area Search Munitions*. MS Thesis, AFIT/GSO/ENY/03-2, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2003.

6. Jacques, David R. "Search, Classification and Attack Decisions for Cooperative Wide Area Search Munitions." *Cooperative Control: Models, Applications and Algorithms*, edited by S. Butenko, et al. 75–93. Kluwer Academic Publishers, 2003.

7. Jacques, David R. and Meir Pachter. "A Theoretical Foundation for Cooperative Search, Classification and Target Attack." *Workshop on Cooperative Control and Optimization.* Gainesville, FL: Kluwer, December 2002.

8. Kish, Brian A. *Establishment of a System Operating Characteristics for Autonomous Wide Area Search Vehicles*. PhD Dissertation, AFIT/DS/ENY/05-5, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 2005.

9. Kish, Brian A., David R. Jacques, and Meir Pachter. "Optimal Control of Sensor Threshold for Autonomous Wide Area Search Munitions," *AIAA Guidance, Naviagtion, and Control Conference and Exhibit*, San Francisco, CA, August 2005.

10. Park, Sang Mork. *Analysis for Cooperative Behavior Effectiveness for Autonomous Wide Area Search Munitions*. MS Thesis, AFIT/GAE/ENY/02-09, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, August 2002

11. Rasmussen, Steven J., Jason W. Mitchell, Phillip R. Chandler, Corey J. Schumacher and Austin L. Smith. *Introduction to the MultiUAV Simulation and Its Application to Cooperative Control Research*. AFRL/VACA, Wright-Patterson AFB OH, June 2005.

12. Rosario, Roland A. *Optimal Sensor Threshold Control and the Weapon Operating Characteristic for Autonomous Search and Attack Munitions*. MS Thesis, AFIT/GAE/ENG/07-02 Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2007

13. Schulz, Christopher S. *Cooperative Control Simulation Validation Using Applied Probability Theory*. MS Thesis, AFIT/GAE/ENY/03S-14, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, August 2003

14. Lockheed Missile and Fire Control Web site: www.missilesandfirecontrol.com/our_products/strikeweapons/LOCAAS/product-locaas.html

Vita

Capt Michael J. Marlin was born in Boulder, Colorado and graduated from Longmont High School in Longmont, Colorado in 1995. He attended the University of Arizona in Tucson, Arizona and earned a Bachelor of Science Degree in Systems Engineering in May 2000. Upon graduation he was commissioned a Second Lieutenant in the United States Air Force. His first assignment in the Air Force was to Eglin AFB, Florida as a munitions test engineer for the 46th Operations Group, Munitions Test Division, Ground Test Branch. While at Eglin AFB, Capt Marlin moved to the 40th Flight Test Squadron to become an F-16 weapons integration test engineer. In June 2003 Capt Marlin moved to Edwards AFB, California to be an operational test engineer in the 31st Test and Evaluation Squadron and test conductor for the B-1B with the Global Power Combined Test Force. In March of 2005 he moved to Wright Patterson AFB, Ohio to attend the Air Force Institute of Technology in pursuit of a Masters of Science Degree in Aeronautical Engineering. While at AFIT Capt Marlin was selected to the USAF Test Pilot School at Edwards AFB, California with a class start date of July 2007.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
15-03-2007	Master's Thesis	March 2005 – March 2007

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
WIDE AREA SEARCH AND ENGAGEMENT SIMULATION VALIDATION	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Marlin, Michael J., Captain, USAF	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	8. PERFORMING ORGANIZATION REPORT NUMBER
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865	AFIT/GAE/ENY/07-M17

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/VAC 2130 Eight St, WPAFB, OH 45433 Dr Jeffrey Tromp (937) 255 3900 AFIT Proposal #2003-120, AFIT JON #05-186	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The use of computer simulation in the development of autonomously controlled unmanned combat aerial vehicles (UCAV) for wide area search and engagement applications is addressed. Computer simulation is an essential tool to analyze control algorithms designed to optimally employ multiple UCAVs in wide area search and engagement. To be representative of real world mission conditions a simulation must be able to accurately duplicate the performance of the automatic target recognition (ATR) methods that will be used to discriminate between targets and non-targets in actual combat. The objective of this research is to demonstrate a method to validate a simulation's ATR model for use in research of wide area search and engagement control schemes. This objective is accomplished by comparing results of multiple simulations of academically contrived wide area search and engagement scenarios to closed form analytic solutions derived for the same scenarios.

15. SUBJECT TERMS: Wide Area Search and Engagement, Wide Area Search Munitions, WASM, UCAV, Autonomous UAV, Automatic Target Recognition, ATR, MultiUAV, UCAV Simulation, Simulation, Cooperative Control, Cooperative Behavior, Cooperative Classification, Cooperative Engagement

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	88	Dr. David R. Jacques
U	U	U			19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3355, ext 3329 (David.Jacques@afit.edu)